Decidable Fragments of First-Order Logic and of First-Order Linear Arithmetic with Uninterpreted Predicates

Marco Voigt

Dissertation

zur Erlangung des Grades
Doktor der Naturwissenschaften (Dr. rer. nat.)
der Fakultät für Mathematik und Informatik
der Universität der Saarlandes

Saarbrücken
Februar 2019

Tag des Kolloquiums: 31. Juli 2019
Dekan: Prof. Dr. Sebastian Hack

Prüfungsausschuss
Vorsitzender: Prof. Dr. Jan Reineke
Berichterstatter: Prof. Dr. Christoph Weidenbach
Prof. Dr. Erich Grädel
Prof. em. Dr. Alexander Leitsch
PD Dr. Thomas Sturm
Akademischer Mitarbeiter: Dr. Benjamin Kiesl

Abstract

First-order logic has a long tradition and is one of the most prominent and most important formalisms in computer science and mathematics. It is well-known that the satisfiability problem for full first-order logic is not solvable algorithmically — we say that first-order logic is undecidable. This fact highlights a fundamental limitation of computing devices in general and of automated reasoning in particular. The classical decision problem, as it is understood today, is the quest for a delineation between the decidable and the undecidable parts of first-order logic based on elegant and computable syntactic criteria. Many researchers have contributed to this endeavor and till today numerous decidable and undecidable fragments of first-order logic have been identified. The present thesis sheds more light on the decidability boundary and aims to open new perspectives on the already known results.

In the first part of the present thesis we focus on the syntactic concept of separateness of variables and explore its applicability to the classical decision problem and beyond. Two disjoint sets of first-order variables are separated in a given formula if each atom in that formula contains variables from at most one of the two sets. This simple notion facilitates the definition of decidable extensions of many well-known decidable first-order fragments. We shall demonstrate that for several prefix fragments, several guarded fragments, the two-variable fragment, and for the fluted fragment. Altogether, we will investigate nine such extensions more closely. Interestingly, each of them contains the monadic first-order fragment without equality. Although the extensions exhibit the same expressive power as the respective originals, certain logical properties can be expressed much more succinctly. In at least two cases the succinctness gap cannot be bounded using any elementary function. This observation can be conceived as an indication for computationally hard satisfiability problems associated with the extended fragments. Indeed, we will derive non-elementary lower bounds for an extension of the Bernays–Schönfinkel–Ramsey fragment, called the separated fragment. Furthermore, we shall investigate the effect of separateness of variables at the semantic level, where it may lead to dependences between quantified variables that are weaker than such dependences are in general. Such weak dependences will be studied in the framework of model-checking games.

The focus of the second part of the present thesis is on linear arithmetic over the rationals with uninterpreted predicates. Two novel decidable fragments shall be presented, both based on the Bernays–Schönfinkel–Ramsey fragment. On the negative side, we will identify several small fragments of the language for which satisfiability is undecidable.

Zusammenfassung

Untersuchungen der Logik erster Stufe blicken auf eine lange Tradition zurück. Es ist allgemein bekannt, dass das zugehörige Erfüllbarkeitsproblem im Allgemeinen nicht algorithmisch gelöst werden kann – man spricht daher von einer unentscheidbaren Logik. Diese Beobachtung wirft ein Schlaglicht auf die prinzipiellen Grenzen der Fähigkeiten von Computern im Allgemeinen aber auch des automatischen Schließens im Besonderen. Das Hilbertsche Entscheidungsproblem wird heute als die Erforschung der Grenze zwischen entscheidbaren und unentscheidbaren Teilen der Logik erster Stufe verstanden, wobei die untersuchten Fragmente der Logik mithilfe klar zu erfassender und berechenbarer syntaktischer Eigenschaften beschrieben werden. Viele Forscher haben bereits zu dieser Untersuchung beigetragen und zahlreiche entscheidbare und unentscheidbare Fragmente entdeckt und erforscht. Die vorliegende Dissertation setzt diese Tradition mit einer Reihe vornehmlich positiver Resultate fort und eröffnet neue Blickwinkel auf eine Reihe von Fragmenten, die im Laufe der letzten einhundert Jahre untersucht wurden.

Im ersten Teil der Arbeit steht das syntaktische Konzept der Separiertheit von Variablen im Mittelpunkt, und dessen Anwendbarkeit auf das Entscheidungsproblem und darüber hinaus wird erforscht. Zwei Mengen von Individuenvariablen gelten bezüglich einer gegebenen Formel als separiert, falls in jedem Atom der Formel die Variablen aus höchstens einer der beiden Mengen vorkommen. Mithilfe dieses leicht verständlichen Begriffs lassen sich viele wohlbekannte entscheidbare Fragmente der Logik erster Stufe zu größeren Klassen von Formeln erweitern, die dennoch entscheidbar sind. Dieser Ansatz wird für neun Fragmente im Detail dargelegt, darunter mehrere Präfix-Fragmente, das Zwei-Variablen-Fragment und sogenannte “guarded” und “fluted” Fragmente. Dabei stellt sich heraus, dass alle erweiterten Fragmente ebenfalls das monadische Fragment erster Stufe ohne Gleichheit enthalten. Obwohl die erweiterte Syntax in den betrachteten Fällen nicht mit einer erhöhten Ausdrucksstärke einhergeht, können bestimmte Zusammenhänge mithilfe der erweiterten Syntax deutlich kürzer formuliert werden. Zumindest in zwei Fällen ist diese Diskrepanz nicht durch eine elementare Funktion zu beschränken. Dies liefert einen ersten Hinweis darauf, dass die algorithmische Lösung des Erfüllbarkeitsproblems für die erweiterten Fragmente mit sehr hohem Rechenaufwand verbunden ist. Tatsächlich wird eine nicht-elementare untere Schranke für den entsprechenden Zeitbedarf beim sogenannten separierten Fragment, einer Erweiterung des bekannten Bernays–Schönfinkel–Ramsey-Fragments, abgeleitet. Darüber hinaus wird der Einfluss der Separiertheit von Individuenvariablen auf der semantischen Ebene untersucht, wo Abhängigkeiten zwischen quantifizierten Variablen durch deren Separiertheit stark abgeschwächt werden können. Für die genauere formale Betrachtung solcher als schwach bezeichneten Abhängigkeiten wird auf sogenannte Hintikka-Spiele zurückgegriffen.

Den Schwerpunkt des zweiten Teils der vorliegenden Arbeit bildet das Entscheidungsproblem für die lineare Arithmetik über den rationalen Zahlen in Verbindung mit uninterpretierten Prädikaten. Es werden zwei bislang unbekannte entscheidbare Fragmente dieser Sprache vorgestellt, die beide auf dem Bernays–Schönfinkel–Ramsey-Fragment aufbauen. Ferner werden neue negative Resultate entwickelt und mehrere unentscheidbare Fragmente vorgestellt, die lediglich einen sehr eingeschränkten Teil der Sprache benötigen.

Contents

Introduction **1**

I Separateness of First-Order Variables **7**

1 Preliminaries **9**

2 Separateness of First-Order Variables **15**

3 Novel Decidable First-Order Fragments **23**
3.1 The Separated Fragment (SF) . . . 28
3.2 Translation of SF into BSR: Upper and Lower Bounds . . . 31
3.3 Expressiveness of SF . . . 39
3.3.1 Fundamental Properties of Relations . . . 39
3.3.2 Basic Counting Quantifiers . . . 40
3.3.3 Expressiveness with Respect to Models of Bounded Size . . . 42
3.4 The Generalized Bernays–Schönfinkel–Ramsey Fragment (GBSR) . . . 55
3.5 Translation of GBSR into BSR . . . 58
3.6 Taking Boolean Structure into Account . . . 61
3.7 The Generalized Ackermann Fragment (GAF) . . . 67
3.8 Translation of GAF into the Ackermann Fragment . . . 70
3.9 The Generalized Gödel–Kalmár–Schütte Fragment (GGKS) . . . 77
3.10 Separateness and Guarded Quantification . . . 82
3.11 Separateness and Guarded Negation . . . 92
3.12 Separateness and Finite-Variable First-Order Logic . . . 95
3.13 Separateness and Fluted Formulas . . . 99
3.14 Decidable Fragments with Function Symbols . . . 102
3.14.1 Unary Functions in Arguments of Monadic Atoms . . . 102
3.14.2 SF and GBSR with Stratified Occurrences of Function Symbols . . . 102
3.14.3 Monadic Horn Sentences in which Positive Literals are Shallow and Linear 104

4 Weak Dependences and Model-Checking Games **111**
4.1 The Simple Case of SF . . . 113
4.2 GBSR Sentences and Uniform Winning Strategies . . . 116
4.3 GAF Sentences and Semi-Uniform Winning Strategies . . . 124

5 Computational Complexity of SF-Sat and GBSR-Sat **139**
5.1 Computational Complexity of Existential SF . . . 142
5.2 Horn and Krom Special Cases of SF and a Conjecture . . . 146
5.3 Proving Lower Bounds for SF-Sat . . . 149
5.3.1 Enforcing a Large Domain in SF . . . 152
5.3.2 Formalizing a Tiling of a Torus . . . 163
5.3.3 Replacing the Equality Predicate . . . 166

6 Interpolation **167**
6.1 Interpolation for SF and GBSR 169
6.2 Interpolation for GAF 174

7 Beyond the Classical Decision Problem **181**
7.1 Separated Formulas and Linear Rational Arithmetic 181
7.2 Skolemization Policies Taking Weak Dependences into Account 191
7.3 Elimination of Second-Order Quantifiers in Second-Order SF 204

II First-Order Linear Arithmetic with Uninterpreted Predicates 215

8 Linear Arithmetic with Uninterpreted Predicates **217**

9 Additional Technical Preliminaries **223**

10 Decidable Fragments of Arithmetic with Uninterpreted Predicates **227**
10.1 Basic Tools from Ramsey Theory 231
10.2 Decidability of BSR with Simple Linear Rational Constraints 233
10.3 BSR(SLR) from the Viewpoint of Combinations of Theories 242
10.4 Decidability of BSR with Bounded Difference Constraints 245
10.5 Formalizing Reachability for Timed Automata in BSR(BD) 254

11 Undecidable Fragments of Arithmetic with Uninterpreted Predicates **263**
11.1 Minsky Machines, Universal Presburger Arithmetic, Simple Encodings 264
11.2 Encoding Two-Counter-Machine Runs in a Unary Predicate 267
11.2.1 Informal Description of the Encoding 268
11.2.2 Formal Encoding of Two-Counter Machine Computations 269
11.2.3 Reducing the Number of Variables to Two 273
11.2.4 Undecidability with One Variable Only Using Another Encoding 274
11.2.5 Using the Rationals or Reals as Underlying Domain 275
11.2.6 Unary Function Symbols and the Horn Fragment 276
11.3 Degrees of Unsolvability 276
11.4 An Encoding Based on Difference Constraints 279
11.4.1 Informal Description of the Encoding 280
11.4.2 Formal Encoding of Two-Counter Machine Computations 281
11.4.3 Restriction to Difference Constraints 284
11.5 Relevance to Verification 285
11.5.1 Separation Logic 286
11.5.2 Verification of Data Structures 287
11.5.3 Verification Using Counter Arithmetic 287
11.5.4 Almost Uninterpreted Formulas with Offsets 288

12 Conclusion **291**
12.1 Separateness: Applications to the Classical Decision Problem and Beyond 291
12.1.1 Applications for the Novel Decidable Fragments 294
12.1.2 More about Future Work 295
12.2 First-Order Linear Arithmetic with Uninterpreted Predicates 296
12.2.1 Applications for the New Decidable Fragments and Future Work 298
12.2.2 Automated Reasoning in Practice: Instantiation Methods 299

List of Figures

1 Overview of known and novel decidable fragments treated in the present thesis . . 3

3.1 Example trees representing integers 47
3.2 Illustration of the structure $\mathcal{F}_{0,1}$. 49
3.3 Nesting of quantifier blocks in the formula $\varphi^{(2n)}$. 60
3.4 Quantifier structure in $\varphi^{(2n)}$ after a first round of narrowing scopes 60
3.5 Quantifier structure in $\varphi^{(2n)}$ after a second round of narrowing scopes 61
3.6 Illustration of the model $\mathcal{A}$ of φ from Example 3.14.9. 108

4.1 Illustration of an exemplary structure $\mathcal{A}$. 125
4.2 Illustration of the structure $\mathcal{C}$ 138

5.1 Computational complexity of subfragments of SF and GBSR 141
5.2 Conjectured computational complexity for Horn and Krom subfragments of SF . . 149

7.1 Solution sets of three arithmetic atoms in two variables 183
7.2 Solution set of three arithmetic atoms in three variables with two fixed values . . . 184
7.3 Solution set of three arithmetic atoms in three variables with one fixed value . . . 184
7.4 Solution set of three arithmetic atoms in three variables without any fixed value . 185
7.5 Solution set of two arithmetic atoms in three variables without any fixed value . . 185

8.1 Partition of the two-dimensional rational plane into equivalence classes 221

10.1 Partition of the two-dimensional rational plane induced by $\sim_{\mathcal{J}_\mathcal{A}}$ 235
10.2 Partitions of the sets $(-2,2)^2$ and $\mathbb{Q}^2$ induced by $\simeq_1$ and $\widehat{\simeq}_1$, respectively 246
10.3 Partition of the set $\mathbb{Q}^2_{\geq 0}$ into $\sim_{\mathfrak{A}}$-equivalence classes 256
10.4 Synchronous versus asynchronous progress of time for a timed automaton 257

11.1 Structure of a single chunk of length $3x$ in a two-counter machine encoding 269
11.2 Structure of a single chunk of length $3d$ in a two-counter machine encoding 281
11.3 Structure of a single chunk of length $3d$ in a two-counter machine encoding 289

12.1 Overview of the novel decidable fragments presented in the present thesis 292
12.2 Partition of the two-dimensional rational plane into equivalence classes 300

List of Tables

1 Summary of the succinctness gaps that are explored in the present thesis 4

5.1 Basic complexity classes and corresponding complete problems 147

11.1 Encoding of two-counter-machine instructions using difference constraints. 266
11.2 The degree of unsolvability regarding certain fragments of Presburger arithmetic with uninterpreted predicates 277
11.3 Encoding of two-counter-machine instructions including a step counter 280

12.1 Summary of the unconditional lower bounds regarding succinctness derived in the present thesis 293
12.2 Summary of the most important undecidability results obtained in Chapter 11 . . 297

Acknowledgments

The present thesis reports on a research endeavor that started in November 2013 when I joined the Automation of Logic group at the Max Planck Institute for Informatics in Saarbrücken, Germany. Along the way my work was influenced by many people — researchers, friends, and family — all of whom I owe a great debt of gratitude. In what follows I would like to take the opportunity to express a small part of that gratitude.

This thesis would not have been possible without the constant support of my thesis supervisor Christoph Weidenbach. He put me on track to the theme of decidability and undecidability in first-order logic and, at the right time, granted me the necessary freedom to develop all the results presented in the thesis. For his generosity and trust in me I am very grateful.

My life at the institute was greatly enriched by my colleagues from the Automation of Logic group and our guests: Gábor Alagi, Noran Azmy, Jasmin Christian Blanchette, Björn Borowski, Aymeric Bouzy, Martin Bromberger, Eugen Denerz, Alberto Fiori, Mathias Fleury, Florian Frohn, Willem Hagemann, Matthias Horbach, Maximilian Jaroschek, Marek Košta, Jennifer Müller, Anna Rossien, Anders Schlichtkrull, Renate Schmidt, Thomas Sturm, Martin Suda, Ching Hoo Tang, Andreas Teucke, Sophie Tourret, Hernán Vanzetto, Daniel Wand, Uwe Waldmann, Christoph Weidenbach, and Patrick Wischnewski. Each and every one of them played their part in turning research work into an enjoyable activity rather than a daily grind. Special thanks go to

- Matthias Horbach for extensive discussions on extensions of first-order rational and Presburger arithmetic with uninterpreted predicates,
- Marek Košta for repeatedly discussing conjunctive associativity and virtual substitution,
- Jennifer Müller for handling even the most troublesome requests most competently, gracefully, and reliably,
- Thomas Sturm for discussing general issues in algebra, separateness of first-order variables, and computational complexity in the context of arithmetic, and for stirring things up every now and again, thereby creating fresh scientific excitement and curiosity,
- Uwe Waldmann for happily discussing all sorts of technical questions in first-order logic and automated reasoning.

Over the years I had discussions with numerous researchers from other institutions who have influenced my work in one way or the other. I am particularly indebted to

- Pascal Fontaine for all the fruitful discussions we have had over the years and, in particular, for pointing out that the BSR(SLR) fragment can be restated in the framework of combinations of theories over non-disjoint vocabularies,
- Erich Grädel for inviting me to contribute to the Algorithmic Model Theory Meeting 2017 and for pointing me to the result by Dawar et al. (2007) and the possible connections of weak dependences to the field of dependence logic,
- Martin Grohe for an inspiring question at LICS 2016 that made me investigate computational complexity of the separated fragment on a fine-grained level,
- Radu Iosif for motivating me to investigate the undecidability boundary for the extension of Presburger arithmetic with uninterpreted predicates,
- Dietrich Kuske for convincing me of the undecidability of BSR(BD) with uninterpreted constants,

Sebastian Rudolph for providing many references to decidability results in knowledge representation,

Viorica Sofronie-Stokkermans for pointing out the similarity between the syntax of BSR(SLR) and the syntax of the array property fragment,

Stanislav Speranski for fruitful discussions about Halpern's (1991) undecidability result concerning Presburger arithmetic with uninterpreted predicates and generalizations of that result,

Christoph Wernhard for inviting me to contribute to the Workshop on Second-Order Quantifier Elimination and Related Topics 2017.

Others who have helped with discussions, comments, advice, or making literature available were Matthias Baaz, Christel Baier, Michael Benedikt, Armin Biere, Maria Paola Bonacina, Uwe Egly, Berit Grußien, Andreas Herzig, Ullrich Hustadt, Yevgeny Kazakov, Manuel Kieroński, Laura Kovacs, Alexander Leitsch, Florian Lonsing, Martin Lück, Stephan Merz, Andreas Nonnengart, Martin Otto, Reinhard Pöschel, Ian Pratt-Hartmann, Karin Quaas, Abhisekh Sankaran, Renate Schmidt, Nicole Schweikardt, Thomas Schwentick, Lidia Tendera, Martin Wirsing, Thomas Zeume, and the anonymous reviewers of papers I contributed to — my apologies go to anyone whom I might have forgotten. For extraordinary cultural contributions I would like to thank Hans de Nivelle for singing so beautifully at the FroCoS 2015 conference dinner and Dexter Kozen for rocking the stage at the LICS 2017 reception. Moreover, I owe thanks to Bernd Finkbeiner who acted as my academic mentor and supported me morally when critical questions arose. I thank the reviewers of the present thesis, Erich Grädel, Alexander Leitsch, Thomas Sturm, and Christoph Weidenbach, for putting all the effort into reviewing more than 300 pages and, moreover, I thank the two additional members of the examination board, Benjamin Kiesl and Jan Reineke, for investing their time.

The teachers who have influenced me most deserve a very special mention: Christel Baier held the lecture *Advaced Logic* at TU Dresden in summer 2011, which has shaped my thinking about first-order logic considerably. I am also very grateful for her very generous support later on. Monika Sturm has taught me the joy of theoretical computer science and the necessity of mathematical rigor. Her excellent lectures on computing formalisms inspired by DNA molecules and unicellular organisms in summer 2008 and on the typed lambda calculus in winter 2008/09 were fascinating and have captured my imagination. It is due to Mrs. Sturm's encouraging and demanding style of teaching and thesis supervision that got me hooked on the field of theoretical computer science.

Finally, I thank my family and friends. Without their unconditional support none of my research activity would have ever happened. I thank my parents and my sister for raising me to become a curios, open-minded, and independent person. To my dear wife, Ina Becher, I owe more than words could ever say for her ever-growing love and care. She cheered me up whenever I was in need and always motivated me to bring things to an end whenever I was in danger of losing focus. I deeply admire her for her patience and courage, and I will always be grateful for her never-ending trust in me.

Marco Voigt

Introduction

What kind of logical reasoning are computing devices capable of?

This is one of the fundamental questions the present thesis ultimately revolves around. In order to provide rigorous answers, we need to be more precise. To this end, we need to, first and foremost, agree on a clear and convincing notion of the involved concepts, namely computing devices and logical reasoning.

There are two major reasons why human-constructed real-world computers and natural objects, such as the human brain, do usually not play much of a role in the study of computing. The first reason is their immense structural and operational complexity, which lies far beyond a reasonable and rigorous mathematical treatment. The second, even more important reason is the ambition of theoretical computer science to obtain results that give interesting insights and that are as broadly applicable as possible. To this end, the used models of computing devices should also capture as much as possible of the computational capabilities of future technology and of yet undiscovered natural phenomena. Therefore, the computing devices we shall take into account in the present thesis are abstract mathematical objects, such as *Turing machines* [Tur36] or *Minsky machines* [Min67]. Although such abstract devices are granted access to potentially unbounded resources, such as running time and information storage capacity, they nevertheless adhere to *finitary principles*: every computation step can only process a fixed finite amount of information, and their behavior can be described by finite means, e.g. by a finite set of rules or by a program of finite length. Both models are characterized by neat and crisp mathematical definitions and they are considered to be *universal models of computation* in the sense that everything that is computable in an intuitive sense can in fact be computed using these abstract computing devices. This is known as the *Church–Turing thesis*.[1] Although it is impossible to prove this claim rigorously, it is widely accepted in contemporary theoretical computer science — similar to laws of nature that are widely accepted in contemporary physics. Indeed, there is strong evidence in favor of the Church–Turing thesis, as all currently known models of computation based on finitary principles turned out to have exactly the same computational power that Turing machines and Minsky machines have. The latter has been proved rigorously, see, e.g., [Min67, HU79, Rog87, Coo04]. What in fact makes the Church–Turing thesis even more interesting is that we today also know that there are inherent limitations on what is computable with Turing machines and all the other equally powerful abstract computing devices. We shall come back to this fact shortly.

The other concept that we need to put on solid grounds is *logical reasoning*. It requires two components: a language in which we can unambiguously formulate the assumptions and propositions we wish to reason about and a finite (better: small) set of plausible rules of inference that describe clearly and unambiguously how conclusions may be drawn from finitely many given assumptions, possibly taking earlier-drawn conclusions into account. Since natural languages are far from being unambiguous, they do not qualify for such an endeavor. The combination of language plus inference rules chosen to be investigated in the present thesis is what is known today as *classical first-order logic*.[2] We will, however, occasionally even consider *classical second-order logic*.

[1] Detailed discussions of the Church–Turing thesis can be found, for instance, in the textbooks [HU79], Sections 7.1 and 7.6, [Rog87], Section 1.7, and in the more recent [Coo04], Chapters 1 and 2, and [Soa16], Chapter 17.

[2] There are many textbooks treating first-order logic on an introductory level, e.g. [End72, Fit96, vD13, Sch08, Smu95, EFT94, TS96].

While Gottfried Wilhelm Leibniz (1646–1716) and David Hilbert (1862–1943) promoted the idea that logical reasoning was more or less fully accessible to computing devices of one kind or another (cf. [Coo04], Chapter 1), Church and Turing showed the contrary in 1936 [Chu36c, Chu36b, Chu36a, Tur36, Tur38]: there are fundamental limits to what computing devices can compute, no matter whether they are abstract mathematical or real-world objects. In particular, it is beyond the capabilities of any abstract computing device, whose behavior is fully describable by a finite set of rules, to always compute the correct answer to the following question for arbitrary finite sets of assumptions S and an arbitrary proposition φ, both formulated in the language of first-order logic: Can we deduce φ from S using only the inference rules admitted in classical first-order logic? Alternatively, one could say that, assuming the Church–Turing thesis to be valid, there is no computing device that can solve the posed question in full generality. This is a limitation that is inherent to each and every model of computing devices that contemporary computer science has to offer.

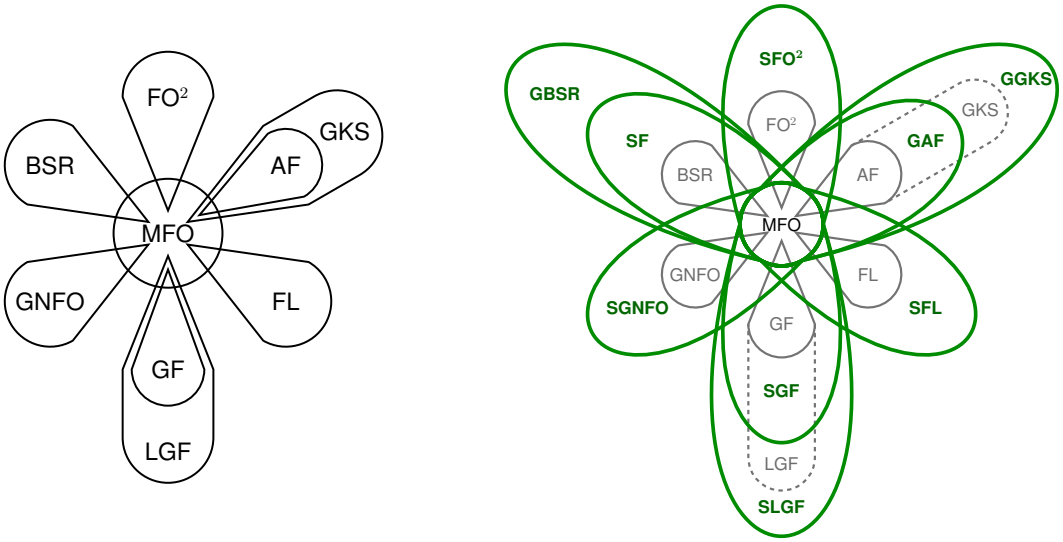
Ever since the discovery of this fundamental insight in the 1930s, computer scientists have constantly put effort into exploring the boundaries between the realm of computable problems and the part that is inaccessible to computing devices. In the field of classical first-order logic, this exploration had even begun before the existence of such a boundary was known. In an attempt to find algorithmic ways for solving the outlined inference problems, several logicians described partial solutions. Instead of tackling the described problem for full first-order logic, they focused on certain parts of the underlying language, that is, on *fragments* of first-order logic, as we shall say, and devised specialized algorithms that are capable of solving instances of the inference problem formulated exclusively in such a language fragment. We shall call such a fragment *decidable* as a reference to Hilbert's *"Entscheidungsproblem"* (German for *decision problem*), which is an equivalent formulation of the inference problem outlined above. After the landmark results by Church and Turing, the exploration started on both sides of the computability boundary and has since then lead to a very large number of discoveries — see Chapter 3 for more details and references. In the present thesis we set out to contribute to this enterprise.

The present thesis is divided into two parts. While Part I concentrates on classical first-order logic without any *background theories*, the focus of Part II is on classical first-order logic enhanced with the language of *linear arithmetic* plus the corresponding rules of reasoning.

In Part I we shall introduce a simple syntactic concept, namely *separateness* of variables, that facilitates elegant definitions of nontrivial extensions of well-known decidable fragments of first-order logic. The extended fragments have the remarkable property of being decidable as well. This aspect of separateness will be fleshed out in Chapter 3 and the novel decidable fragments presented therein constitute one of the main contributions of the present thesis. Figure 1 provides a schematic overview of the most important fragments we shall introduce. Roughly speaking, variables are separated in a first-order formula, if they never co-occur in the basic building blocks of the formula, called *atoms*. From a qualitative point of view, the extended fragments do not come with an increased expressiveness compared to the original fragments. That is, every property that can be expressed in such an extended language can also be expressed in the underlying original language. However, in some cases the respective logical formulas in the original fragment may have to be much longer. Indeed, we will show for several cases that there are significant gaps regarding the length of shortest formulas that express one and the same property, formulated in the extended language on the one hand, compared to the formulation in the original language on the other hand. For example, the succinctness gap between the well-known *Bernays–Schönfinkel–Ramsey fragment* and an extension of it, which we shall call the *separated fragment*, is as follows. For every positive natural number n we can find some property that can be expressed in the separated fragment with a formula of length $k \cdot n^2$ for some positive natural number k, whereas expressing the very same property in the Bernays–Schönfinkel–Ramsey fragment requires a formula whose length is at least

$$2^{2^{\cdot^{\cdot^{\cdot^{2^2}}}}} \Big\} \text{height } n \ .$$

Hence, from the perspective of formula length, the extended fragments we shall present enable



MFO	– monadic first-order fragment	SF	– separated fragment
BSR	– Bernays–Schönfinkel–Ramsey fragment	GBSR	– generalized BSR
FO^2	– two-variable fragment	SFO^2	– separated FO^2
AF	– Ackermann fragment	GAF	– generalized AF
GKS	– Gödel–Kalmár–Schütte fragment	GGKS	– generalized GKS
FL	– fluted fragment	SFL	– separated FL
GF	– guarded fragment	SGF	– separated GF
LGF	– loosely guarded fragment	SLGF	– separated LGF
GNFO	– guarded negation fragment	SGNFO	– separated GNFO

Figure 1: Left-hand side: Schematic overview of well-known decidable fragments of first-order logic. Only the partial overlaps between MFO and the other fragments is depicted. We neglect any other partial overlaps. Moreover, the containment of AF in GKS and of GF in LGF is shown. Right-hand side: Schematic overview of the extended fragments (in green) that shall be presented in Chapter 3. Notice that MFO is properly contained in all extended fragments. The focus is again on the overlaps with MFO and on the proper containment relations. Other depicted overlaps might be unsubstantiated.

us to describe certain properties much more succinctly and elegantly. Table 1 summarizes the succinctness gaps that we shall derive.

This boost in succinctness comes with a price tag attached concerning the *computational complexity* of logical inference in the extended fragments. That is, a worst-case analysis of the resources consumed by an abstract computing device that is actually doing logical reasoning in the separated fragment rather than in the Bernays–Schönfinkel–Ramsey fragment, for instance, reveals a similar gap regarding the required running time. It turns out that logical reasoning in the separated fragment is computationally as hard as logical reasoning is in any decidable first-order fragment that enjoys the so-called *finite model property*, if the size of smallest models is bounded by linearly growing towers of exponentials. This in fact means that logical reasoning for the separated fragment is at least as time consuming as it is for any other decidable first-order fragment enjoying the finite model property that is known today — at least as far as the author of the present thesis is aware of. We shall show this in two different ways: once in Chapter 3 (Theorem 3.3.11) — relative to the reasoning problem associated with other decidable first-order fragments — and once in Chapter 5 (Theorem 5.3.11) — relative to computationally hard problems from a different domain, so-called *domino problems*.

More succinct fragment	Less succinct fragment	Succinctness gap	Reference
SF	BSR	non-elementary	Theorem 3.2.7
SF	Gaifman-local first-order fragment	non-elementary	Theorem 3.3.18
GAF	AF	super-polynomial	Proposition 3.8.9
GGKS	GKS	exponential	Theorem 3.9.9
SGF	LGF	non-elementary	Theorem 3.10.8
SFO^2	FO^2	exponential	Theorem 3.12.5

Table 1: Summary of the succinctness gaps the are explored in the present thesis. The abbreviations for fragments are spelled out in Figure 1. The first row, for instance, summarizes the succinctness gap between the separated fragment and the Bernays–Schönfinkel–Ramsey fragment that we have described above. The gap between GAF and AF is conditional on ExpTime $\neq$ NExpTime. All other gaps are unconditional.

Beyond the already described aspects, the concept of separateness of variables has further interesting facets on offer to be explored. A gentle introduction to the technical aspects of separateness and a detailed overview of the related topics treated in the present thesis can be found in Chapter 2. The syntactic notion of separateness induces the following semantic counterpart. Under certain circumstances, two first-order variables, one *universally quantified* and the other *existentially quantified*, that are separated in a given formula φ show a dependence pattern that is weaker than the dependence pattern one encounters in general in first-order logic. Hence, the term *weak dependence* suggests itself. In the general case, if a universally quantified variable x ranges over infinitely many values, an existentially quantified variable y that depends on x may have to range over infinitely many values as well. However, if y depends only weakly on x and the values assigned to all other involved variables are fixed, then the range of y can always be restricted to a finite set of values. This phenomenon and some consequences will be investigated in detail in Chapter 4. It will also play a role in Chapter 7, Section 7.2, where we shall suggest how *Skolemization*, an important technique in *automated reasoning*, could be enhanced so as to make it sensitive to weak dependences and yield better outcomes. Another interesting topic that is worth mentioning is *interpolation*. A first-order fragment is said to be closed under interpolation, if for every formula φ_1 from the fragment that *logically entails* another formula φ_2 from the fragment there is a third formula ψ from the same fragment, called the *interpolant of* φ_1 *and* φ_2 that "sits between" φ_1 and φ_2 in a syntactic and semantic sense. This means that (a) the vocabulary that is used in ψ is the common vocabulary of φ_1 and φ_2, and (b) φ_1 logically entails ψ and ψ logically entails φ_2. We shall derive interpolation theorems for several of our novel decidable first-order fragments: the *separated fragment* and the *generalized Bernays–Schönfinkel–Ramsey fragment* (Theorem 6.1.1), the *generalized Ackermann fragment* (Theorem 6.2.1), and the *separated guarded negation fragment* (Proposition 6.0.4).

In Part II of the present thesis, we turn our attention to a more specific part of classical first-order logic, which is concisely described as *first-order arithmetic with uninterpreted predicate symbols*. Logical reasoning in first-order arithmetic is not fully accessible to computing devices either. This changes, however, if we restrict multiplication. For instance, logical reasoning in the logic of the integers with addition, equality, and strict order — a fragment known as *Presburger arithmetic* — can be done by computing devices without human interaction, at least in principle. The picture changes again, as soon as we add so-called *uninterpreted predicate symbols* to the language. Allowing such predicate symbols, even if they have only one argument place, yields a logic fragment that is not decidable (cf. Chapter 11). Again, we need to restrict the admitted language, in order to obtain a decidable fragment.

Indeed, the main purpose of Part II is to explore the boundary between what parts of logical reasoning is accessible to computing devices in first-order arithmetic with uninterpreted predicate symbols and the parts that are inaccessible to computing devices. A detailed introduction and

overview can be found in Chapter 8. In Chapter 10 we shall present two positive results, and in Chapter 11 several negative results will be derived. The two decidable fragments introduced in Chapter 10 are the *Bernays–Schönfinkel–Ramsey fragment with simple linear rational constraints* and the *Bernays–Schönfinkel–Ramsey fragment with bounded difference constraints*. Both fragments are suitable for applications in hardware and software verification.[3] For instance, we shall discuss in Section 10.5 how the latter fragment can be used to verify safety properties of real-time systems. In Chapter 11 the positive results shall be contrasted with an investigation of the syntactic threshold of the computationally inaccessible part of logical reasoning. We will concentrate on restricted fragments of Presburger arithmetic plus uninterpreted predicate symbols. Most of these results can be easily transfered to arithmetic over the rational numbers. In particular, we shall show in Section 11.4 how small relaxations of the syntactic requirements characterizing the Bernays–Schönfinkel–Ramsey fragment with bounded difference constraints will turn this decidable fragment into a fragment where logical reasoning is not fully accessible to computing devices anymore. Finally, in Section 11.5 we will sketch what the negative results found in Chapter 11 mean for certain formalisms used in the filed of verification, in particular concerning limitations that are revealed by our discoveries.

The following list summarizes the main contributions of the present thesis:

(1) The as yet unexplored concept of separateness of variables is fleshed out in several directions (Part I, an overview is given in Chapter 2).

(2) Nine novel decidable fragments of first-order logic are introduced that extend well-known decidable first-order fragments (Chapter 3). The major fragments that are being introduced are the *separated fragment (SF)*, the *generalized Bernays–Schönfinkel–Ramsey fragment (GBSR)*, the *generalized Ackermann fragment (GAF)*, the *generalized Gödel–Kalmár–Schütte fragment (GGKS)*, the *separated guarded fragment (SGF)*, the *separated loosely guarded fragment (SLGF)*, the *separated guarded-negation fragment (SGNFO)*, the *separated two-variable fragment (SFO^2)*, and the *separated fluted fragment (SFL)*, cf. Figure 1. Moreover, it is proved that the qualitative expressiveness of the extended fragments compared to the respective original fragments stays the same.

(3) Significant gaps regarding succinctness are derived for several of the extended fragments: SF, GBSR, GAF, GGKS, SGF, SLGF, SFO^2, cf. Table 1. This evidently shows that several of the extended fragments constitute a substantial quantitative improvement regarding expressiveness compared to the original fragments. Moreover, a succinctness gap is shown between SF sentences and shortest equivalent Gaifman-local sentences (Theorem 3.3.18).

(4) As a semantic counterpart to the mostly syntactically-minded investigation of separateness, the notion of *weak dependence* is introduced and it is investigated in the framework of *model-checking games* and *satisfying strategies* (also: *winning strategies*) in the spirit of Hintikka (Chapter 4). It is shown that every first-order sentence in which all dependences are weak is equivalent to some Bernays–Schönfinkel–Ramsey sentence (Theorem 4.2.1).

(5) Regarding the *computational complexity* of the *satisfiability problem* for SF (SF-Sat) and GBSR (GBSR-Sat), it is shown that both problems are non-elementary by deriving upper and lower bounds (Chapter 5). More precisely, it is shown that both problems are Tower-complete (cf. [Sch16], see also Definition 5.0.2) and that for every positive integer k there are k-NExpTime-complete subproblems in both SF-Sat and GBSR-Sat (Theorem 5.0.3). Furthermore, a polynomial-time reduction is devised that facilitates reducing the satisfiability problem of any first-order fragment enjoying the *finite model property* to SF-Sat, provided that the former fragment comes with an elementary (or small non-elementary) bound regarding the size of smallest models (Theorem 3.3.11).

[3] In fact, BSR(SLR) and BSR(BD) can be conceived as generalizations of formalisms that are already used for verification, such as the Bernays–Schönfinkel–Ramsey fragment itself, the existential fragment of linear arithmetic, or *difference constraints* — references are given in Section 12.1.1, in Chapter 8 (Remarks 8.0.1 and 8.0.2 and the subsection on related work), and in Sections 11.5 and 12.2.1.

(6) Craig–Lyndon-style interpolation theorems are proved for the Bernays–Schönfinkel–Ramsey fragment, SF, GBSR, the Ackermann fragment, and GAF, all without equality (Theorem 6.1.1, Lemma 6.1.9, Theorem 6.2.1, and Lemma 6.2.5).

(7) Several applications of separateness beyond the definition of decidable first-order fragments are proposed and first promising results derived: new *Skolemization* techniques sensitive to *weak dependences* (Section 7.2), a second-order variant of SF admitting the elimination of certain second-order quantifiers (Section 7.3), and an analysis of separateness in the context of interpreted logics, e.g. *linear rational arithmetic* (Section 7.1).

(8) Novel fragments of first-order linear rational arithmetic enhanced with uninterpreted predicate symbols are defined and proved to be decidable (Chapter 10): the *Bernays–Schönfinkel–Ramsey fragment with simple linear rational constraints (BSR(SLR))* and the *Bernays–Schönfinkel–Ramsey fragment with bounded difference constraints (BSR(BD))*. It is shown that both fragments have a NExpTime-complete satisfiability problem (Corollaries 10.2.15 and 10.4.11). The decidability proof for BSR(SLR) can be restated in the framework of *combinations of theories*, which facilitates extensions, e.g. based on SF or GBSR rather than the Bernays–Schönfinkel–Ramsey fragment, and based on polynomials rather than linear arithmetic terms only (Section 10.3). A non-trivial application of BSR(BD) is elaborated upon, namely *reachability analysis* for formal models of *real-time systems*, in particular for *timed automata* (Section 10.5).

(9) Finally, several undecidable fragments of first-order arithmetic with uninterpreted predicate symbols are identified and discussed, mostly based on the universal fragment of *Presburger arithmetic* or the universal fragment of *linear rational arithmetic* (or restricted subfragments thereof) with a single uninterpreted predicate symbol of arity one (Chapter 11).

Parts of the material developed in the present thesis have been published in conference proceedings [SVW16, Voi17b, Voi17a, HVW17a], workshop proceedings [Voi17d], and as preprint [VW15, Voi17c, HVW17b]. Parts of the texts in the present thesis were taken, adapted, and extended from these papers without explicitly giving references to the respective source. All of the thus used parts were originally written by the author of the present thesis.

Part I

Separateness of First-Order Variables: A New Viewpoint on the Classical Decision Problem and Beyond

Chapter 1

Preliminaries

Syntax of First-Order Formulas

We mainly consider first-order logic formulas with equality. The following notions and notation are fairly standard and can be found in different composition in standard texts about first-order logic, for example, [End72, CK90, EFT94, Smu95, Fit96, vD13].

A *vocabulary* $\Sigma = \langle \Pi, \Omega \rangle$ (also: *signature*) comprises a countable set Π of predicate symbols and a countable set Ω of function symbols. If not explicitly stated otherwise, the vocabularies treated in the present thesis are finite. Every symbol in Σ is equipped with a nonnegative integer, its *arity*. For the distinguished *equality* predicate, whose semantics is fixed to be the identity relation, we use $\approx$. A function symbol of arity zero is called *constant symbol*. We call a vocabulary $\Sigma = \langle \Pi, \Omega \rangle$ *relational* if Ω is empty.

vocabulary $\Sigma = \langle \Pi, \Omega \rangle$

Fix some vocabulary $\Sigma = \langle \Pi, \Omega \rangle$ and fix some countably infinite supply Var of first-order variables. A Σ-*term* is a finite syntactic object: any constant symbol $c \in \Omega$ is a Σ-term; any variable $v \in \mathrm{Var}$ is a Σ-term; given any m-ary function symbol $f \in \Omega$ and m Σ-terms $t_1, \ldots, t_m$ the expression $f(t_1, \ldots, t_m)$ is also a Σ-term. *Atomic* Σ-*formulas* (also: Σ-*atoms*) are either the *logical constants* `true`, `false`, or are *equations* $s_1 \approx s_2$, or are of the form $P(s_1, \ldots, s_m)$ where the s_i are Σ-terms, P stems from Π, and m is the arity of P. A Σ-*formula* is either a Σ-atom, a *negated* Σ-*formula* $\neg\varphi$, a *conjunction* $\varphi \wedge \psi$, a *disjunction* $\varphi \vee \psi$, an *implication* $\varphi \to \psi$, an *equivalence* $\varphi \leftrightarrow \psi$, or a *quantified* Σ-*formula* of the form $\forall x.\, \varphi$ or $\exists y.\, \varphi$, where φ, ψ are Σ-formulas and $x, y \in \mathrm{Var}$ are first-order variables. For any Σ-terms s, t we use $s \not\approx t$ to abbreviate $\neg\, s \approx t$. Given a quantified Σ-formula $\mathcal{Q}v.\, \varphi$, we call the subformula φ the *scope* of the quantifier $\mathcal{Q}v$. Similarly, in a negated Σ-formula $\neg\varphi$ the subformula φ is the *scope* of this occurrence of the negation sign. In order to save parentheses, we follow the convention that negation binds strongest, that conjunction binds stronger than disjunction, and that all of the aforementioned bind stronger than implication and equivalence. Equivalence, in turn, binds weaker than implication. The scope of quantifiers shall stretch as far to the right as admitted by parentheses. Given a set Φ of Σ-formulas, we call a Σ-formula φ a *Boolean combination of formulas from* Φ, if φ consist of formulas from Φ, possibly connected via the Boolean connectives $\neg, \wedge, \vee, \to, \leftrightarrow$. If we restrict the set of Boolean connectives even further to $\wedge, \vee$, we speak of a $\wedge$-$\vee$-*combination of formulas from* Φ. Given some finite sequence of pairwise distinct variables $v_1, \ldots, v_n$, the expression $\mathcal{Q}v_1 \ldots v_n.\, \varphi$ with $\mathcal{Q} \in \{\forall, \exists\}$ abbreviates the formula $\mathcal{Q}v_1.\mathcal{Q}v_2. \ldots \mathcal{Q}v_n.\, \varphi$. In addition, we often use the tuple notation $\mathcal{Q}\bar{\mathrm{v}}$ for the same purpose, where $\bar{\mathrm{v}}$ stands for any finite tuple $\langle v_1, \ldots, v_n \rangle$ with pairwise distinct first-order variables. For convenience, we often identify tuples $\bar{\mathrm{v}}$ of variables with the set containing all the variables that occur in $\bar{\mathrm{v}}$. In most cases this abstraction from the exact order of quantifiers is justified by Proposition 1.0.1. A *quantifier block* is a maximal sequence $\mathcal{Q}v_1.\mathcal{Q}v_2. \ldots \mathcal{Q}v_n$ of quantifiers of the same kind occurring in a given formula. We occasionally use regular expressions to describe sequences of quantifiers. For example, for any positive integer k the expression $\exists^*\forall^k\exists\exists$ stands for the set of all prefixes of the form $\exists y_1 \ldots y_m \forall x_1 \ldots x_k \exists z_1 z_2$, where m ranges over all nonnegative integers; in particular, the leading existential quantifier block may be empty. Oftentimes the

Var

Boolean combination

vocabulary underlying our considerations shall not be mentioned explicitly. We just speak of *terms*, *formulas*, and *atoms*, when Σ is not important or clear from the current context. A formula is called *relational*, if the underlying vocabulary is relational, i.e. it does not contain any function symbols.

relational formula

A variable v occurs *freely* in a formula φ if the formula contains an occurrence of v that is not in the scope of any quantifier $\mathcal{Q}v$ in φ. An occurrence of a variable v in a formula φ is *bound*, if it lies within the scope of some quantifier $\mathcal{Q}v$ in φ. In all formulas we tacitly assume, if not explicitly stated otherwise, that no variable occurs freely and bound at the same time and that all distinct occurrences of quantifiers bind distinct variables. We use $\varphi(v_1, \ldots, v_m)$ to denote a formula φ whose free first-order variables form a subset of $\{v_1, \ldots, v_m\}$. The variables $v_1, \ldots, v_m$ are assumed to be pairwise distinct. A formula is *closed* if it does not contain any free occurrences of variables. A closed formula is also called a *(Σ-)sentence*. We call a term or a formula *ground* if it does not contain any occurrences of variables, neither free nor bound, and no quantifiers.

$\varphi(v_1, \ldots, v_m)$

We denote *substitution* by $\varphi\big[v/s\big]$, where every free occurrence of v in φ is to be substituted by the term s. For *simultaneous substitution* of pairwise distinct variables $v_1, \ldots, v_n$ with $s_1, \ldots, s_n$, respectively, we use the notation $\varphi\big[v_1/s_1, \ldots, v_n/s_n\big]$. For example, $P(x, y)\big[x/f(y), y/g(x)\big]$ results in $P(f(y), g(x))$. Notice that this is different from the sequential application of substitution in $P(x, y)\big[x/f(y)\big]\big[y/g(x)\big] = P\big(f(g(x)), g(x)\big)$ and $P(x, y)\big[y/g(x)\big]\big[x/f(y)\big] = P\big(f(y), g(f(y))\big)$. We also write $\big[\bar{v}/\bar{s}\big]$ to abbreviate $\big[v_1/s_1, \ldots, v_n/s_n\big]$.

$\varphi\big[v_1/s_1, \ldots, v_n/s_n\big]$

A formula is in *prenex normal form* if it has the shape $\mathcal{Q}_1 v_1 \ldots \mathcal{Q}_n v_n.\, \psi$ with quantifier-free ψ and $\mathcal{Q}_i \in \{\forall, \exists\}$, i.e. all quantifiers are lined up in front of the formula. The quantifier-free part is sometimes referred to as *matrix*. A formula is in *negation normal form* if it exclusively contains the connectives $\wedge, \vee, \neg$ and every negation sign occurs immediately in front of an atom; quantifiers are of course admitted. A *literal* is an atom or a negated atom, and a *clause* is a disjunction of literals. A *unit clause* is a clause containing exactly one literal. We say that a formula is in *conjunctive normal form (CNF)* if it is a conjunction of clauses, possibly preceded by a quantifier prefix.[1] A formula in CNF is *Horn* if every clause contains at most one non-negated literal. It is *Krom* if every clause contains at most two literals. A formula is in *disjunctive normal form (DNF)* if it is a disjunction of conjunctions of literals, possibly preceded by a quantifier prefix. A sentence $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n.\, \psi$ is said to be in *standard form* if ψ is quantifier free and in negation normal form, in particular, the connectives $\rightarrow$ and $\leftrightarrow$ are not admitted; all variables bound in the quantifier prefix are required to actually occur in ψ. The tuples $\bar{\mathrm{x}}_1$ and $\bar{\mathrm{y}}_n$ may be empty, i.e. the quantifier prefix does not have to start with a universal quantifier, and it does not have to end with an existential quantifier.

normal forms

standard form

For any formula φ we denote by $\mathrm{vars}(\varphi)$ the set of all variables occurring freely or bound in φ. Moreover, we write $\mathrm{consts}(\varphi)$ to address the set of all constant symbols that occur in φ. Similar notation is used for other syntactic objects.

$\mathrm{vars}(\varphi)$, $\mathrm{consts}(\varphi)$

Next, we define a measure of length of terms and formulas. We set $\mathrm{len}(c) := 1$ and $\mathrm{len}(v) := 1$ for every constant symbol c and every variable v. For terms $s_1, \ldots, s_m$ we set $\mathrm{len}\big(f(s_1, \ldots, s_m)\big) := 1 + \sum_{i=1}^{m} \mathrm{len}(s_i)$. The logical constants `true` and `false` are assigned length 1. The length of the other atoms is given by $\mathrm{len}\big(P(s_1, \ldots, s_m)\big) := 1 + \sum_{i=1}^{m} \mathrm{len}(s_i)$, which includes the case where P is the equality predicate. For formulas φ, ψ we set $\mathrm{len}(\neg\varphi) := 1 + \mathrm{len}(\varphi)$, $\mathrm{len}(\varphi \wedge \psi) = \mathrm{len}(\varphi \vee \psi) := 1 + \mathrm{len}(\varphi) + \mathrm{len}(\psi)$, $\mathrm{len}(\forall \bar{\mathrm{x}}.\, \psi) := 1 + |\bar{\mathrm{x}}| + \mathrm{len}(\psi)$, and $\mathrm{len}(\exists \bar{\mathrm{y}}.\, \psi) := 1 + |\bar{\mathrm{y}}| + \mathrm{len}(\psi)$. Moreover, in the context of the length of formulas, we conceive implication and equivalence as abbreviations and set $\mathrm{len}(\varphi \rightarrow \psi) := \mathrm{len}(\neg\varphi \vee \psi)$ and $\mathrm{len}(\varphi \leftrightarrow \psi) := \mathrm{len}\big((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)\big)$. Notice that, in the presence of the connective $\leftrightarrow$, $\mathrm{len}(\varphi)$ can be exponentially greater than the number of symbols needed to write φ down.

length of terms and formulas

The *quantifier rank* of a formula is the depth of quantifier nestings in the formula. Every quantifier-free formula has quantifier rank zero. The quantifier rank of any formula $\mathcal{Q}v.\, \psi$ with $\mathcal{Q} \in \{\forall, \exists\}$ is the quantifier rank of ψ plus one. For every formula $\varphi \circ \psi$ with $\circ \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ the quantifier rank is the maximum of the quantifier ranks of φ and ψ. Every $\neg\varphi$ has the same

quantifier rank

[1] In contrast to the tradition in automated reasoning, we do allow quantifier prefixes in formulas in conjunctive or disjunctive normal form.

quantifier rank as φ has.

For every Σ-formula φ with $\Sigma = \langle \Pi, \Omega \rangle$ we define $\|\varphi\|$ to be the *length of some fixed encoding of* φ in a binary alphabet, e.g. suitable for the tape of some Turing machine. We assume the encoding to be *reasonable*, i.e. we assume that $\|\varphi\| \in \mathcal{O}\big(\mathrm{len}(\varphi) \cdot \log\big(|\Pi| + |\Omega| + |\mathrm{vars}(\varphi)|\big)\big)$ if the underlying vocabulary $\Sigma = \langle \Pi, \Omega \rangle$ is finite. $\|\varphi\|$

Semantics of First-Order Formulas

We mainly follow the Tarskian approach to semantics and interpret logic formulas with respect to given structures. Let $\Sigma := \langle \Pi, \Omega \rangle$ be a vocabulary. A Σ-*structure* $\mathcal{A}$ consists of a nonempty set A, its *domain* (also: *universe*), and *interpretations* $f^{\mathcal{A}}$ and $P^{\mathcal{A}}$ of all function and predicate symbols in Σ. More precisely, $\mathcal{A}$ interprets any m-ary function symbol $f \in \Omega$ by a total mapping $f^{\mathcal{A}} : \mathsf{A}^m \to \mathsf{A}$ that maps each and every m-tuple of elements from A to some element from A. Moreover, $\mathcal{A}$ interprets any m-ary predicate symbol $P \in \Pi$ by a (possibly empty) set $P^{\mathcal{A}} \subseteq \mathsf{A}^m$ of m-tuples over $\mathcal{A}$'s domain. Like for terms and formulas, we mostly omit the explicit reference to the underlying vocabulary Σ when speaking about structures. *structure*

Given any structure $\mathcal{A}$, a *variable assignment* (over $\mathcal{A}$'s domain) is a total mapping $\beta : \mathrm{Var} \to \mathsf{A}$ that assigns domain elements $\mathsf{a} \in \mathsf{A}$ to variables $v \in \mathrm{Var}$. When the domain A is clear from the context we often do not explicitly specify it, e.g. when a particular structure $\mathcal{A}$ is discussed, variable assignments are implicitly understood to assign elements from A, if not explicitly stated otherwise. We sometimes explicitly define a variable assignment by writing $[v_1 \mapsto \mathsf{a}_1, \ldots, v_m \mapsto \mathsf{a}_m]$ — or $[\bar{\mathrm{v}} \mapsto \bar{\mathsf{a}}]$ for short — when it is not important which elements are assigned to the variables in $\mathrm{Var} \setminus \{v_1, \ldots, v_m\}$. In such cases, we consider just some variable assignment β with $\beta(v_i) = \mathsf{a}_i$ for $i = 1, \ldots, m$. Given some variable assignment β, we define its *update* $\beta[v_1 \mapsto \mathsf{a}_1, \ldots, v_m \mapsto \mathsf{a}_m]$ to be the variable assignment β' with $\beta'(v_i) = \mathsf{a}_i$ for $i = 1, \ldots, m$ and $\beta'(v') = \beta(v')$ for every $v' \in \mathrm{Var} \setminus \{v_1, \ldots, v_m\}$. *variable assignment* $\beta[\bar{\mathrm{v}} \mapsto \bar{\mathsf{a}}]$

Given a Σ-term s, a Σ-structure $\mathcal{A}$, and a variable assignment β over $\mathcal{A}$'s domain, we denote the *evaluation of* s *under* $\mathcal{A}$ *and* β by $\mathcal{A}(\beta)(s)$. It is defined such that $\mathcal{A}(\beta)(v) := \beta(v)$ for variables v, $\mathcal{A}(\beta)(c) := c^{\mathcal{A}}$ for constant symbols c, and $\mathcal{A}(\beta)\big(f(s_1, \ldots, s_m)\big) := f^{\mathcal{A}}\big(\mathcal{A}(\beta)(s_1), \ldots, \mathcal{A}(\beta)(s_m)\big)$ for complex terms. As for every ground term s the value of s under $\mathcal{A}$ is independent of any variable assignment, we occasionally drop the reference to any specific variable assignment and simply write $\mathcal{A}(s)$ when s is ground. Given a Σ-formula φ, a Σ-structure $\mathcal{A}$, and a variable assignment β over $\mathcal{A}$'s domain, we say that φ is *satisfied under* $\mathcal{A}$ *and* β, written $\mathcal{A}, \beta \models \varphi$, if the following conditions are met: We always have $\mathcal{A}, \beta \models \texttt{true}$ but never $\mathcal{A}, \beta \models \texttt{false}$. For more complicated formulas φ we define the following: $\mathcal{A}(\beta)(s)$ $\mathcal{A}, \beta \models \varphi$

$\mathcal{A}, \beta \models s \approx t$	if and only if	$\mathcal{A}(\beta)(s) = \mathcal{A}(\beta)(t)$,
$\mathcal{A}, \beta \models P(s_1, \ldots, s_m)$	if and only if	$\big\langle \mathcal{A}(\beta)(s_1), \ldots, \mathcal{A}(\beta)(s_m) \big\rangle \in P^{\mathcal{A}}$,
$\mathcal{A}, \beta \models \neg\psi$	if and only if	$\mathcal{A}, \beta \models \psi$ does not hold,
$\mathcal{A}, \beta \models \psi \wedge \chi$	if and only if	$\mathcal{A}, \beta \models \psi$ and $\mathcal{A}, \beta \models \chi$,
$\mathcal{A}, \beta \models \psi \vee \chi$	if and only if	$\mathcal{A}, \beta \models \psi$ or $\mathcal{A}, \beta \models \chi$,
$\mathcal{A}, \beta \models \psi \to \chi$	if and only if	$\mathcal{A}, \beta \models \psi$ implies $\mathcal{A}, \beta \models \chi$,
$\mathcal{A}, \beta \models \psi \leftrightarrow \chi$	if and only if	$\mathcal{A}, \beta \models \psi$ implies $\mathcal{A}, \beta \models \chi$ and vice versa,
$\mathcal{A}, \beta \models \forall x.\, \psi$	if and only if	$\mathcal{A}, \beta[x \mapsto \mathsf{a}] \models \psi$ for every $\mathsf{a} \in \mathsf{A}$,
$\mathcal{A}, \beta \models \exists y.\, \psi$	if and only if	$\mathcal{A}, \beta[y \mapsto \mathsf{b}] \models \psi$ for some $\mathsf{b} \in \mathsf{A}$.

If φ is not satisfied under $\mathcal{A}$ and β, we write $\mathcal{A}, \beta \not\models \varphi$. When there is no danger of confusion, we sometimes conveniently abbreviate expressions of the form $\mathcal{A}, [v_1 \mapsto \mathsf{a}_1, \ldots, v_m \mapsto \mathsf{a}_m] \models \varphi(v_1, \ldots, v_m)$ by $\mathcal{A} \models \varphi(\mathsf{a}_1, \ldots, \mathsf{a}_m)$. We write $\mathcal{A} \models \varphi$ if $\mathcal{A}, \beta \models \varphi$ holds for every variable assignment β over $\mathcal{A}$'s domain. In such cases, we say that $\mathcal{A}$ is a *model* of φ. For sentences φ we often omit the variable assignment and say that $\mathcal{A}$ *satisfies* φ if $\mathcal{A}, \beta \models \varphi$ for any β. A sentence φ is called *satisfiable* if it has a model, i.e. if there is some structure $\mathcal{A}$ with $\mathcal{A} \models \varphi$. Otherwise, we call φ *unsatisfiable* or *inconsistent*. Two sentences φ and ψ are considered *equisatisfiable* if φ has a model if and only if ψ has one. Furthermore, a sentence φ is called *valid*, if it is satisfied under any structure; if there is at least one structure not satisfying φ, then the sentence if *invalid*. A sentence is *valid with respect*

to a certain class of structures, if it is satisfied under every structure from the class. In particular in Part II of the present thesis we shall occasionally use this latter notion of *validity with respect to a class of structures* without explicitly referring to the class of structures, if it is clear from the current context.

We also use the symbol $\models$ to denote *semantic entailment* of two formulas (over the same vocabulary). A formula φ *semantically entails* a formula ψ, written $\varphi \models \psi$, whenever for every structure $\mathcal{A}$ and every variable assignment β, $\mathcal{A}, \beta \models \varphi$ implies $\mathcal{A}, \beta \models \psi$. The symbol $\models\!\!\!\models$ denotes *semantic equivalence* of formulas, i.e. $\varphi \models\!\!\!\models \psi$ holds whenever $\varphi \models \psi$ and $\psi \models \varphi$. For convenience, we often drop the word "semantic" and just speak of entailment and equivalence.

$\varphi \models\!\!\!\models \psi$

A *logical* Σ*-theory* is a set of Σ-sentences closed under semantic entailment, i.e. for every sentence φ with $\mathcal{T} \models \varphi$ we have $\varphi \in \mathcal{T}$. Given a logical theory $\mathcal{T}$, two formulas $\varphi(\bar{\mathrm{x}}), \psi(\bar{\mathrm{x}})$ are considered $\mathcal{T}$*-equivalent*, if $\mathcal{T} \models \forall \bar{\mathrm{x}}.\, \varphi(\bar{\mathrm{x}}) \leftrightarrow \psi(\bar{\mathrm{x}})$. For any Σ-structure $\mathcal{A}$ we write $\mathrm{Th}(\mathcal{A})$ to address the logical theory of all Σ-sentences that are satisfied by $\mathcal{A}$. It is easy to check that $\mathrm{Th}(\mathcal{A})$ is closed under semantic entailment. For convenience, we sometimes use the term $\mathcal{A}$-equivalence when we actually mean $\mathrm{Th}(\mathcal{A})$-equivalence.

$\mathcal{T}$*-equivalence*

We shall often use the fact that quantifiers can be *shifted* in certain ways within formulas under preservation of the formula's semantics. The following proposition collects the basic equivalences that facilitate shifting of quantifiers.

Proposition 1.0.1 (Quantifier shifting)**.** *Let φ, ψ, χ be formulas, and assume that x and y do not occur freely in χ. We have the following equivalences, where $\circ \in \{\wedge, \vee\}$:*

(i)	$\exists y.\, (\varphi \vee \psi)$	$\models\!\!\!\models$	$(\exists y.\, \varphi) \vee (\exists y.\, \psi)$	*(ii)*	$\forall x.\, (\varphi \wedge \psi)$	$\models\!\!\!\models$	$(\forall x.\, \varphi) \wedge (\forall x.\, \psi)$
(iii)	$\exists y.\, (\varphi \circ \chi)$	$\models\!\!\!\models$	$(\exists y.\, \varphi) \circ \chi$	*(iv)*	$\forall x.\, (\varphi \circ \chi)$	$\models\!\!\!\models$	$(\forall x.\, \varphi) \circ \chi$
(v)	$\exists y_1 \exists y_2.\, \varphi$	$\models\!\!\!\models$	$\exists y_2 \exists y_1.\, \varphi$	*(vi)*	$\forall x_1 \forall x_2.\, \varphi$	$\models\!\!\!\models$	$\forall x_2 \forall x_1.\, \varphi$

Consequently, if $x_1 \notin \mathrm{vars}(\chi)$ and $x_2 \notin \mathrm{vars}(\varphi)$ holds for two first-order formulas φ and χ, we get $(\exists x_1.\, \varphi) \wedge (\exists x_2.\, \chi) \models\!\!\!\models \exists x_1 x_2.\, (\varphi \wedge \chi)$ and dually $(\forall x_1.\, \varphi) \vee (\forall x_2.\, \chi) \models\!\!\!\models \forall x_1 x_2.\, (\varphi \vee \chi)$.

Consider any first-order sentence φ that contains an occurrence of a subformula ψ of the form $\exists y.\, \psi'$ and let $\mathcal{Q}_1 u_1, \ldots, \mathcal{Q}_n u_n$ be the sequence of all quantifiers from φ in whose scope ψ lies. Let $\bar{\mathrm{x}}$ be the tuple containing all the variables from the list $v_1, \ldots, v_n$ that are universally quantified in φ. The process of replacing the occurrence of ψ in φ with the formula $\psi'\big[y/f_y(\bar{\mathrm{x}})\big]$ for some fresh function symbol f_y of appropriate arity is called *(standard) Skolemization*[2] of y (or of $\exists y$); the term $f_y(\bar{\mathrm{x}})$ is called *Skolem term* and the function symbol f_y *Skolem function*. In case of $|\bar{\mathrm{x}}| = 0$, we call f_y *Skolem constant*. By *exhaustive Skolemization* of a given sentence φ we mean Skolemization of all the existential first-order quantifiers in φ one after the other (in any order).

Skolemization

Proposition 1.0.2. *Let φ be some first-order Σ-sentence and let φ' be the result of Skolemizing some of the existential quantifiers in φ. Then, we observe $\varphi' \models \varphi$ and any model $\mathcal{A}$ of φ can be turned into a model $\mathcal{B}$ of φ' by extending $\mathcal{A}$ with appropriate interpretations of the introduced Skolem functions and Skolem constants. The rest of $\mathcal{A}$ remains unchanged, i.e. $\mathcal{A}$ and $\mathcal{B}$ coincide with respect to their domains and their interpretations of the symbols in Σ.*

A structure $\mathcal{A}$ is a *substructure* of a structure $\mathcal{B}$ (over the same vocabulary) if (1) $\mathsf{A} \subseteq \mathsf{B}$, (2) $c^{\mathcal{A}} = c^{\mathcal{B}}$ for every constant symbol c, (3) $P^{\mathcal{A}} = P^{\mathcal{B}} \cap \mathsf{A}^m$ for every m-ary predicate symbol P, and (4) $f^{\mathcal{A}}(\bar{\mathsf{a}}) = f^{\mathcal{B}}(\bar{\mathsf{a}})$ for every m-ary function symbol f and every m-tuple $\bar{\mathsf{a}} \in \mathsf{A}^m$. Given a structure $\mathcal{A}$ and some subset S of $\mathcal{A}$'s domain, the *substructure of $\mathcal{A}$ induced by S* is the unique substructure $\mathcal{B}$ of $\mathcal{A}$ with the domain $\mathsf{B} := S$. The following is a standard lemma from model theory.

substructure

Lemma 1.0.3 (Substructure Lemma)**.** *Let φ be a first-order sentence without existential quantifiers and in which no universal quantifier lies within the scope of any negation sign — we treat any subformula $\varphi_1 \to \varphi_2$ as abbreviation for $\neg\varphi_1 \vee \varphi_2$ and any subformula $\varphi_1 \leftrightarrow \varphi_2$ as abbreviation for $(\neg\varphi_1 \vee \varphi_2) \wedge (\varphi_1 \vee \neg\varphi_2)$ to account for implicit negation signs as well. Moreover, let $\mathcal{A}$ be a substructure of $\mathcal{B}$. If $\mathcal{B} \models \varphi$, then $\mathcal{A} \models \varphi$.*

[2] Notice that the form of *standard Skolemization* used in the present thesis is regarded as being inefficient in automated reasoning, as there are improved variants available that produce Skolem terms with fewer arguments. We shall discuss this topic in more depth in Section 7.2.

Proof (inspired by [EFT94], proof of Lemma 5.7 in Chapter III).
It is easy to prove by induction on a term s that

($*$) for every variable assignment β over $\mathcal{A}$'s domain we have $\mathcal{A}(\beta)(s) = \mathcal{B}(\beta)(s)$.

Now let ψ be any first-order formula. By induction on the structure of ψ, we show that for every variable assignment β over $\mathcal{A}$'s domain we have that $\mathcal{B}, \beta \models \psi$ entails $\mathcal{A}, \beta \models \psi$. If ψ is atomic, the claim follows from ($*$). If ψ is of the form $\neg\chi$, $\chi \wedge \eta$, $\chi \vee \eta$, $\chi \rightarrow \eta$, or $\chi \leftrightarrow \eta$, the claim follows easily from the inductive hypothesis. Suppose ψ is of the form $\forall x.\, \chi$. Further assume that $\mathcal{B}, \beta[x \mapsto \mathsf{a}] \models \chi$ holds for every $\mathsf{a} \in \mathsf{B}$. Then, by induction, we have $\mathcal{A}, \beta[x \mapsto \mathsf{a}] \models \chi$ for every $\mathsf{a} \in \mathsf{A} \subseteq \mathsf{B}$. In other words, we have $\mathcal{A}, \beta \models \forall x.\, \chi$.

This finishes the proof of the claim. The Substructure Lemma follows as a corollary. □

Herbrand structure

A special kind of structures are *Herbrand structures*.[3] The domain of a Herbrand Σ-structure $\mathcal{H}$ — called *Herbrand domain* — is the set of all ground Σ-terms, where we assume that Σ contains at least one constant symbol. Moreover, every m-ary function symbol $f \in \Omega$ is interpreted by $\mathcal{H}$ such that for any sequence $t_1, \ldots, t_m \in \mathsf{H}$ we have $f^{\mathcal{H}}(t_1, \ldots, t_m) = f(t_1, \ldots, t_m)$. This means that $\mathcal{H}$'s domain and the interpretation of function symbols under $\mathcal{H}$ are entirely determined by the underlying vocabulary. Solely the interpretation of predicate symbols can (and must) be chosen. The following is a standard lemma which justifies that in certain contexts it suffices to exclusively consider Herbrand structures.

Lemma 1.0.4. *Let φ be a sentence without equality and without existential quantifiers and in which no universal quantifier lies within the scope of any negation sign. If φ is satisfiable, then it has a model that is a Herbrand structure.*

Proof. Let $\mathcal{A}$ be any model of φ and let Σ be the vocabulary underlying φ. Let $\mathcal{A}'$ be the substructure of $\mathcal{A}$ that is induced by the domain

$$\mathsf{A}' := \left\{ \mathsf{a} \in \mathsf{A} \mid \text{there is a ground } \Sigma\text{-term } t \text{ such that } \mathsf{a} = \mathcal{A}(t) \right\} .$$

By the Substructure Lemma, $\mathcal{A}'$ is also a model of φ. We define the Herbrand structure $\mathcal{H}$ such that for every m-ary predicate symbol $P \in \Pi$ and for all ground Σ-terms $t_1, \ldots, t_m$ we set

$$\langle t_1, \ldots, t_m \rangle \in P^{\mathcal{H}} \qquad \text{if and only if} \qquad \left\langle \mathcal{A}'(t_1), \ldots, \mathcal{A}'(t_m) \right\rangle \in P^{\mathcal{A}'} .$$

Let γ be any variable assignment over the Herbrand domain H and let β be the variable assignment defined such that $\beta(v) := \mathcal{A}(\gamma(v))$ for every v. Then, we observe for every non-equational Σ-atom A that $\mathcal{H}, \gamma \models A$ if and only if $\mathcal{A}', \beta \models A$. As φ contains only universal quantifiers that do not lie within the scope of any negation sign, this observation together with $\mathcal{A} \models \varphi$ entails $\mathcal{H} \models \varphi$. □

Given some Σ-sentence φ, a Herbrand Σ-model $\mathcal{H} \models \varphi$ is called *minimal*, if there is no Herbrand Σ-model $\mathcal{H}' \models \varphi$ such that for every predicate symbol P in Σ we have $P^{\mathcal{H}'} \subseteq P^{\mathcal{H}}$ and one of these inclusions is strict. The following proposition is a standard result, see, e.g. Theorem 3.8 in Chapter XI of [EFT94].

Proposition 1.0.5. *Every satisfiable first-order Σ-sentence φ that is Horn has a* unique *minimal Herbrand model (sometimes also called the* least *Herbrand model). In other words, there is some Herbrand model $\mathcal{H}_* \models \varphi$ such that for every Herbrand model $\mathcal{H}$ of φ we have $P^{\mathcal{H}_*} \subseteq P^{\mathcal{H}}$ for every predicate symbol P in Σ.*

[3] In automated reasoning Herbrand structures are often represented by sets of atoms over a given vocabulary. We deviate from this definition, although the intended semantical object is ultimately the same.

Syntax and Semantics of Second-Order Formulas

Occasionally, we shall also consider second-order formulas with equality. That is, we add second-order quantification $\forall P.\,\psi$ and $\exists P.\,\psi$ to the inductive syntax definition of formulas, where ψ is a second-order formula and P is some predicate symbol that may occur in ψ but does not have to. Similarly, we add quantifiers $\forall f.\,\psi$ and $\exists f.\,\psi$ for any function symbol f. Like for first-order variables, we define free and bound occurrences of second-order variables for predicate and function symbols. Moreover, we also tacitly assume for second-order formulas that no variable occurs free and bound in the same formula and that distinct occurrences of quantifiers bind distinct variables, if not explicitly stated otherwise. For semantic satisfaction under a given structure $\mathcal{A}$ and a given variable assignment β we use the following rules in addition to the above first-order conditions:

$$\begin{aligned}
&\mathcal{A},\beta \models \forall P.\,\psi \quad \text{if and only if} \quad \mathcal{A}',\beta \models \psi \text{ for every structure } \mathcal{A}' \text{ that differs from } \mathcal{A} \text{ only in the interpretation of } P,\\
&\mathcal{A},\beta \models \exists P.\,\psi \quad \text{if and only if} \quad \mathcal{A}',\beta \models \psi \text{ for some structure } \mathcal{A}' \text{ that differs from } \mathcal{A} \text{ only in the interpretation of } P,\\
&\mathcal{A},\beta \models \forall f.\,\psi \quad \text{if and only if} \quad \mathcal{A}',\beta \models \psi \text{ for every structure } \mathcal{A}' \text{ that differs from } \mathcal{A} \text{ only in the interpretation of } f,\\
&\mathcal{A},\beta \models \exists f.\,\psi \quad \text{if and only if} \quad \mathcal{A}',\beta \models \psi \text{ for some structure } \mathcal{A}' \text{ that differs from } \mathcal{A} \text{ only in the interpretation of } f,
\end{aligned}$$

where ψ is any second-order formula. The notions and notation for semantic entailment and semantic equivalence are extended to second-order formulas in the obvious way.

Additional Notation

$[k]$

We use the notation $[k]$ to abbreviate the set $\{1,\ldots,k\}$ for any positive integer k. The *power set* of a set S, i.e. the set of all subsets of S, is denoted by $\mathcal{P}(S)$. The iterated application of $\mathcal{P}$ is given by $\mathcal{P}^0(S) := S$ and $\mathcal{P}^{k+1}(S) := \mathcal{P}^k(\mathcal{P}(S))$ for $k \geq 0$. For convenience, we mostly drop the parentheses and simply write $\mathcal{P}^k S$. Furthermore, we also define the *tetration operation* inductively by $2^{\uparrow 0}(m) := m$ and $2^{\uparrow k+1}(m) := 2^{\left(2^{\uparrow k}(m)\right)}$.

$\mathcal{P}^k S$

$2^{\uparrow k}(m)$

$S/_\sim$

Let S be any nonempty set. For any equivalence relation $\sim \subseteq S \times S$ we write $S/_\sim$ to address the set $\left\{S' \subseteq S \;\middle|\; S' \text{ is a maximal nonempty set such that for all } a,b \in S' \text{ we have } a \sim b\right\}$, which we shall call the *quotient set* (or simply *quotient*) induced by $\sim$ over S. The sets in $S/_\sim$ are the *equivalence classes* induced by $\sim$ over S. Given any element $a \in S$, we write $[a]_\sim$ to address the (unique) equivalence class containing a. Given any two equivalence relations $\sim_1, \sim_2 \subseteq S \times S$, we call $\sim_1$ a *refinement* of $\sim_2$ if (a) for every set $T \in S/_{\sim_1}$ there is some set $T' \in S/_{\sim_2}$ such that $T \subseteq T'$, and (b) for every set $T' \in S/_{\sim_2}$ there is a finite collection of sets $T_1,\ldots,T_k \in S/_{\sim_1}$ such that $T' = T_1 \cup \ldots \cup T_k$.

$[a]_\sim$

refinement

Chapter 2

Separateness of First-Order Variables

We now introduce a fairly simple concept that shall be the key theme in the entire Part I of the present thesis: *separateness of first-order variables.*

Definition 2.0.1 (Separateness of first-order variables)**.** *Let φ be any first-order formula and let X, Y be two disjoint sets of first-order variables. We say that X and Y are* separated in φ *if for every atom A occurring in φ we have* $\mathrm{vars}(A) \cap X = \emptyset$ *or* $\mathrm{vars}(A) \cap Y = \emptyset$ *or both. We say that X, Y are* strictly separated in φ *if X and Y are separated in φ and, in addition, for every subformula $\chi := (\mathcal{Q}v.\ \ldots)$ of φ we either have* $\mathrm{vars}(\chi) \cap X = \emptyset$ *or* $\mathrm{vars}(\chi) \cap Y = \emptyset$.

Intuitively speaking, two sets X, Y of variables are separated in a formula, if there are no co-occurrences of variables $x \in X$ and $y \in Y$ in any atom. This simple syntactic notion is the key to a number of results that we shall develop in the subsequent chapters. Examples are novel decidable fragments of first-order logic (Chapter 3), computationally hard satisfiability problems in first-order logic (Chapter 5), and new insights regarding the dependences between universally and existentially quantified first-order variables belonging to separated sets, with an application to Skolemization (Chapters 4 and 7). In the rest of the present chapter we give a more detailed overview of separateness and its applications.

Typically, one would expect that sets of first-order variables in "naturally occurring" formulas are either not separated or are separated in trivial ways. For example, consider a formula that stipulates that R is a strict ordering without endpoint:

$$\begin{aligned}\varphi := \ & \big(\forall x_1 y_1 z_1.\, R(x_1, y_1) \wedge R(y_1, z_1) \rightarrow R(x_1, z_1)\big) \\ & \wedge \big(\forall x_2 y_2.\, R(x_2, y_2) \rightarrow \neg R(y_2, x_2)\big) \\ & \wedge \big(\forall x_3 \exists y_3.\, R(x_3, y_3)\big)\ .\end{aligned}$$

In any of the three conjuncts all occurring variables co-occur in some atom. On the other hand, the three sets $\{x_1, y_1, z_1\}, \{x_2, y_2\}, \{x_3, y_3\}$ are pairwise separated in φ. This trivial kind of separation is due to the fact that φ is simply a conjunction of three closed formulas.

Non-trivial cases of separateness appear, for instance, in formulas where universal and existential quantifiers are nested and the variables they bind are separated. Consider the sentence $\psi := \forall x \exists y.\, P(x) \leftrightarrow Q(y)$ in which the singleton sets $\{x\}$ and $\{y\}$ are obviously separated. It expresses a certain symmetry in structures $\mathcal{A}$. For every domain element a there is some element b such that a belongs to $P^{\mathcal{A}}$ if and only if b belongs to $Q^{\mathcal{A}}$. It turns out that the same property can be expressed without any nesting of alternating quantifiers. Indeed, we can use the distributivity laws of Boolean algebra and quantifier shifting (cf. Proposition 1.0.1) to transform ψ into the equivalent

sentence $\big((\exists x.\, P(x)) \to (\exists y_1.\, Q(y_1))\big) \wedge \big((\exists x.\, \neg P(x)) \to (\exists y_2.\, \neg Q(y_2))\big)$:

$$\begin{aligned}
&\forall x \exists y.\, P(x) \leftrightarrow Q(y) \\
&\models\!\!\models\ \forall x \exists y.\, \big(\neg P(x) \vee Q(y)\big) \wedge \big(P(x) \vee \neg Q(y)\big) \\
&\models\!\!\models\ \forall x.\, \big(\neg P(x) \wedge (\exists y_2.\, \neg Q(y_2))\big) \vee \big((\exists y_1.\, Q(y_1)) \wedge P(x)\big) \\
&\models\!\!\models\ \big((\forall x.\, \neg P(x)) \vee (\exists y_1.\, Q(y_1))\big) \wedge \big((\exists y_2.\, \neg Q(y_2)) \vee (\forall x.\, P(x))\big) \\
&\models\!\!\models\ \big((\exists x.\, P(x)) \to (\exists y_1.\, Q(y_1))\big) \wedge \big((\exists x.\, \neg P(x)) \to (\exists y_2.\, \neg Q(y_2))\big)
\end{aligned}$$

We could even shift quantifiers outwards again and finally obtain an equivalent sentence with a $\exists\exists\forall$ quantifier prefix: $\psi' := \exists y_1 y_2 \forall x.\, \big(P(x) \to Q(y_1)\big) \wedge \big(\neg P(x) \to \neg Q(y_2)\big)$. This example shows that we can not only transform nested quantification of separated variables into quantification that is not nested. In addition, we can replace $\forall\exists$ alternations in exchange for $\exists\forall$ alternations, or vice versa.

In the example we start from a $\forall\exists$ sentence and obtain an $\exists\exists\forall$ sentence. The increase in the number of used quantifiers is not a coincidence. Much rather, it illustrates a key difference between the two representations. The sentence ψ can, using a $\forall\exists$ alternation, represent the symmetry property of structures more succinctly than the sentence ψ' can with an $\exists\forall$ quantifier alternation. The following examples illustrates this phenomenon in a more pronounced way. We shall see later that such succinctness gaps can become k-fold exponential, if we start from k nested $\forall\exists$ alternations and seek an equivalent sentence with a single $\exists\forall$ quantifier alternation. The proof of this result (cf. Theorem 3.2.7) is based on a general variant of ψ.

Example 2.0.2. *Consider the sentence $\varphi_1 := \forall x \exists y.\, (P_1(x) \leftrightarrow Q_1(y)) \wedge \ldots \wedge (P_n(x) \leftrightarrow Q_n(y))$. Given any sequence $\bar{b} := b_1 \ldots b_n$ of n bits, we denote by $\chi_{\bar{b}}(x)$ and $\eta_{\bar{b}}(y)$ the formulas*

$$\chi_{\bar{b}}(x) := \bigwedge_{\substack{1 \le i \le n \\ b_i = 1}} P_i(x) \quad \wedge \bigwedge_{\substack{1 \le j \le n \\ b_j = 0}} \neg P_j(x) \qquad \text{and} \qquad \eta_{\bar{b}}(y) := \bigwedge_{\substack{1 \le i \le n \\ b_i = 1}} Q_i(y) \quad \wedge \bigwedge_{\substack{1 \le j \le n \\ b_j = 0}} \neg Q_j(y)\ .$$

Then, φ_1 can be transformed into the equivalent sentence

$$\forall x. \bigvee_{\bar{b} \in \{0,1\}^n} \chi_{\bar{b}}(x) \wedge \exists y.\, \eta_{\bar{b}}(y)\ ,$$

where we have managed to shift the existential quantifier $\exists y$ inwards. We can do the same for the universal quantifier $\forall x$, if we beforehand transform the sentence into a conjunction of disjunctions. To keep the sentence short, we do not just blindly apply the Boolean laws of distributivity, but we also remove redundant formula parts — we have already tacitly done so in the above transformation. This results in the sentence

$$\bigwedge_{\bar{b} \in \{0,1\}^n} \big(\forall x.\, \neg \chi_{\bar{b}}(x)\big) \vee \exists y.\, \eta_{\bar{b}}(y) \qquad \models\!\!\models \qquad \bigwedge_{\bar{b} \in \{0,1\}^n} \big(\exists x.\, \chi_{\bar{b}}(x)\big) \to \exists y.\, \eta_{\bar{b}}(y)\ .$$

We can now shift quantifiers outwards again, existential ones first. Since existential quantification does not distribute over conjunction, we have to rename bound existential variables. We thus obtain the equivalent sentence

$$\varphi_1' := \exists \underbrace{y_{0\ldots0} \ldots y_{1\ldots1}}_{2^n \text{ variables}} \forall x \bigwedge_{\bar{b} \in \{0,1\}^n} \chi_{\bar{b}}(x) \to \eta_{\bar{b}}(y_{\bar{b}})\ .$$

The sentence φ_1' is much more verbose that the original φ_1. The original φ_1 refers to the universal variable x and stipulates the existence of a counterpart y that behaves with respect to Q like x behaves with respect to P. In contrast, the sentence φ_1' lists 2^n elements, including detailed descriptions of their potential behavior with respect to Q, and stipulates the existence of each and every single one of them, provided a counterpart exhibiting the respective behavior with respect to P is contained in

the domain. Indeed, no equivalent sentence in prenex form with the quantifier prefix $\exists^*\forall^*$ could do significantly better (cf. Theorem 3.2.7).

Now consider the case with additional quantifier alternations:

$$\varphi_2 := \forall u \exists v \forall x \exists y.\, (P_1(u,x) \leftrightarrow Q_1(v,y)) \wedge \ldots \wedge (P_n(u,x) \leftrightarrow Q_n(v,y)) \ .$$

We extend the notation $\chi_{\bar{b}}(x)$ from above to $\chi_{\bar{b}}(u,x)$ by replacing every $P_i(x)$ with $P_i(u,x)$ and every $\neg P_j(x)$ with $\neg P_j(v,y)$. In the same spirit the notation $\eta_{\bar{b}}(y)$ is extended to $\eta_{\bar{b}}(v,y)$. When we apply the transformations from above to the new formula in one step, we obtain

$$\forall u \exists v. \bigwedge_{\bar{b} \in \{0,1\}^n} \big(\forall x. \neg \chi_{\bar{b}}(u,x)\big) \vee \exists y. \eta_{\bar{b}}(v,y) \ .$$

We next transform the scope of $\exists v$ into a disjunction of conjunctions and, while doing so, treat the subformulas $\forall x. \neg \chi_{\bar{b}}(u,x)$ and $\exists y. \eta_{\bar{b}}(v,y)$ as indivisible units. Shifting the quantifier $\exists v$ inwards then yields

$$\forall u. \bigvee_{S \subseteq \{0,1\}^n} \Big(\Big(\bigwedge_{\bar{b} \in \{0,1^n\} \setminus S} \forall x. \neg \chi_{\bar{b}}(u,x) \Big) \wedge \exists v. \bigwedge_{\bar{b} \in S} \exists y. \eta_{\bar{b}}(v,y) \Big) \ .$$

Next, we use the distributivity laws to transform the scope of $\forall u$ into a conjunction of disjunctions, in order to be able to shift the quantifier inwards:

$$\begin{aligned} &\bigwedge_{S \subseteq \{0,1\}^n} \Big(\Big(\forall u. \bigvee_{\bar{b} \in S} \forall x. \neg \chi_{\bar{b}}(u,x) \Big) \vee \exists v. \bigwedge_{\bar{b} \in S} \exists y. \eta_{\bar{b}}(v,y) \Big) \\ &\models \bigwedge_{S \subseteq \{0,1\}^n} \Big(\Big(\exists u. \bigwedge_{\bar{b} \in S} \exists x. \chi_{\bar{b}}(u,x) \Big) \rightarrow \exists v. \bigwedge_{\bar{b} \in S} \exists y. \eta_{\bar{b}}(v,y) \Big) \ . \end{aligned}$$

Finally, we can shift all quantifiers to the front again and thus obtain the sentence

$$\varphi_2' := \exists v_{S_1} \ldots v_{S_m} \exists \bar{y}_{S_1} \ldots \bar{y}_{S_m} \forall u \forall \bar{x}. \bigwedge_{S \subseteq \{0,1\}^n} \Big(\Big(\bigwedge_{\bar{b} \in S} \chi_{\bar{b}}(u, x_{\bar{b}}) \Big) \rightarrow \bigwedge_{\bar{b} \in S} \eta_{\bar{b}}(v_S, y_{S,\bar{b}}) \Big) \ ,$$

where $S_1, \ldots, S_m$ is an enumeration of all subsets of $\{0,1\}^n$, i.e. $m = 2^{2^n}$; each $\bar{y}_{S_i}$ is a tuple of $|S_i|$ variables $y_{S_i,\bar{b}}$ with $\bar{b} \in S_i$; and $\bar{x}$ is a tuple of 2^n variables $x_{\bar{b}}$ with $\bar{b} \in \{0,1\}^n$.

Again, the sentence φ_2' is much more verbose than the original φ_2. This time the gap in succinctness grows even doubly exponential with growing n. And, once more, no equivalent $\exists^*\forall^*$-sentence could, asymptotically speaking, do much better.

The examples we have seen so far illustrate a general property of nested quantification of variables that are separated. Namely, nesting of quantifiers is not essential but may facilitate a more succinct representation of properties. The formula transformations in Example 2.0.2 illustrate how such succinct representations can be unfolded. We can use the same approach to prove the following technical lemma.

Lemma 2.0.3. *Let $\bar{x}, \bar{y}, \bar{x}', \bar{y}', \bar{z}$ be pairwise disjoint tuples of first-order variables and let $\psi(\bar{x}, \bar{x}', \bar{y}, \bar{y}', \bar{z})$ be a formula in which $\bar{x} \cup \bar{x}'$ and $\bar{y} \cup \bar{y}'$ are strictly separated. We can transform $\forall \bar{x} \exists \bar{y}.\, \psi(\bar{x}, \bar{x}', \bar{y}, \bar{y}', \bar{z})$ into an equivalent formula $\psi'(\bar{x}', \bar{y}', \bar{z})$ that satisfies the following conditions.*

- *(a) The sets $\bar{x} \cup \bar{x}'$ and $\bar{y} \cup \bar{y}'$ are strictly separated in ψ'.*
- *(b) The quantifier alternation caused by the $\forall^*\exists^*$ prefix in $\forall \bar{x} \exists \bar{y}.\, \psi$ vanishes in ψ'. More precisely, for any subformula χ in ψ' of the form $\big(\mathcal{Q}u. \ldots (\mathcal{Q}'v. \ldots) \ldots\big)$ with $\mathcal{Q} \neq \mathcal{Q}'$*
  - *either χ entirely stems from $\psi(\bar{x}, \bar{x}', \bar{y}, \bar{y}', \bar{z})$ (modulo renaming of bound variables),*
  - *or $(\mathcal{Q}'v. \ldots)$ stems from $\psi(\bar{x}, \bar{x}', \bar{y}, \bar{y}', \bar{z})$ and the quantifier $\mathcal{Q}u$ stems from the prefix $\forall \bar{x} \exists \bar{y}$ (modulo renaming of bound variables).*

basic formulas

Proof. A *basic formula* is any atom and any subformula $(\mathcal{Q}v.\ldots)$ in ψ that does not lie within the scope of any quantifier in ψ.

We first transform ψ into an equivalent disjunction of conjunctions of negated or non-negated basic formulas. This is always possible. Since the sets $\bar{\mathrm{x}} \cup \bar{\mathrm{x}}'$ and $\bar{\mathrm{y}} \cup \bar{\mathrm{y}}'$ are strictly separated in ψ, none of the basic formulas contains variables from both sets. Hence, the constituents of every conjunction can be grouped into three parts: $\psi_i(\bar{\mathrm{x}}, \bar{\mathrm{x}}', \bar{\mathrm{z}})$, containing none of the variables from $\bar{\mathrm{y}} \cup \bar{\mathrm{y}}'$; $\chi_i(\bar{\mathrm{y}}, \bar{\mathrm{y}}', \bar{\mathrm{z}})$, containing none of the variables from $\bar{\mathrm{x}} \cup \bar{\mathrm{x}}'$; $\eta_i(\bar{\mathrm{x}}', \bar{\mathrm{y}}', \bar{\mathrm{z}})$, containing neither variables from $\bar{\mathrm{x}}$ nor from $\bar{\mathrm{y}}$. Hence, $\forall\bar{\mathrm{x}}\exists\bar{\mathrm{y}}.\, \psi(\bar{\mathrm{x}}, \bar{\mathrm{x}}', \bar{\mathrm{y}}, \bar{\mathrm{y}}', \bar{\mathrm{z}})$ is equivalent to a formula of the form

$$\forall\bar{\mathrm{x}}\exists\bar{\mathrm{y}}.\, \bigvee_i \psi_i(\bar{\mathrm{x}}, \bar{\mathrm{x}}', \bar{\mathrm{z}}) \wedge \chi_i(\bar{\mathrm{y}}, \bar{\mathrm{y}}', \bar{\mathrm{z}}) \wedge \eta_i(\bar{\mathrm{x}}', \bar{\mathrm{y}}', \bar{\mathrm{z}}) \ ,$$

where the ψ_i, χ_i, and η_i are conjunctions of negated or non-negated basic formulas. We now shift the existential quantifier block $\exists\bar{\mathrm{y}}$ inwards so that it only binds the (sub-)conjunctions $\chi_i(\bar{\mathrm{y}}, \bar{\mathrm{y}}', \bar{\mathrm{z}})$. The emerging subformulas $\big(\exists\bar{\mathrm{y}}.\, \chi_i(\bar{\mathrm{y}}, \bar{\mathrm{y}}', \bar{\mathrm{z}})\big)$ are treated as basic formulas in the further process (replacing the ones that now occur as their proper subformulas). Notice that the sets $\bar{\mathrm{x}} \cup \bar{\mathrm{x}}'$ and $\bar{\mathrm{y}} \cup \bar{\mathrm{y}}'$ are still strictly separated in the scope of the leading $\forall\bar{\mathrm{x}}$ quantifier block.

Next, we transform the formula into a conjunction of disjunctions of negated and non-negated basic formulas, group the constituents of disjunctions into two groups $\psi'_j(\bar{\mathrm{x}}, \bar{\mathrm{x}}', \bar{\mathrm{z}})$ and $\chi'_j(\bar{\mathrm{x}}', \bar{\mathrm{y}}', \bar{\mathrm{z}})$, and shift the universal quantifier block $\forall\bar{\mathrm{x}}$ inwards so that it only binds the (sub-)disjunctions $\psi'_j(\bar{\mathrm{x}}, \bar{\mathrm{x}}', \bar{\mathrm{z}})$. The resulting formula is the sought ψ' in which the sets $\bar{\mathrm{x}} \cup \bar{\mathrm{x}}'$ and $\bar{\mathrm{y}} \cup \bar{\mathrm{y}}'$ are still strictly separated. □

We shall apply variants of Lemma 2.0.3 and the general methods used in its proof in several places, mostly for resolving $\forall\exists$ quantifier alternations. But the underlying idea is much more general. If certain separateness conditions are satisfied by a formula, succinct representations of properties can be unfolded into more verbose ones that require a lower quantifier rank or even use fewer quantifier alternations. To this end, notice that the prefix $\forall\bar{\mathrm{x}}\exists\bar{\mathrm{y}}$ in Lemma 2.0.3 could be replaced with a $\exists^*\forall^*$ prefix or any other prefix over the variables in $\bar{\mathrm{x}}$ and $\bar{\mathrm{y}}$. It is straightforward to adapt the proof to the new situation. We only need to reorder the steps of the quantifier shifting scheme and/or more iterations.

The following result can easily be proven using Lemma 2.0.3.

Lemma 2.0.4. *Let $\varphi(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n, \bar{\mathrm{z}})$ be a quantifier-free formula in which the sets $\bar{\mathrm{x}} := \bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_n$ and $\bar{\mathrm{y}} := \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_n$ are separated. There exists a quantifier-free formula $\varphi'(\bar{\mathrm{u}}, \bar{\mathrm{v}}, \bar{\mathrm{z}})$ such that $\forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1 \ldots \forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n.\, \varphi(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n, \bar{\mathrm{z}})$ and $\exists\bar{\mathrm{u}}\forall\bar{\mathrm{v}}.\, \varphi'(\bar{\mathrm{u}}, \bar{\mathrm{v}}, \bar{\mathrm{z}})$ are equivalent.*

Proof. For quantifier-free formulas separateness and strict separateness coincide. Hence, we can apply Lemma 2.0.3 to the formula $\forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1 \ldots \forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n.\, \varphi(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n, \bar{\mathrm{z}})$ in an iterated fashion to obtain some equivalent formula ψ in which the sets $\bar{\mathrm{x}}$ and $\bar{\mathrm{y}}$ are strictly separated. As none of the alternations in the quantifier prefix $\forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1 \ldots \forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n$ is present in ψ, for any of its subformulas $\big(\mathcal{Q}u.\ldots(\mathcal{Q}'v.\ldots)\ldots\big)$ with $\mathcal{Q}, \mathcal{Q}' \in \{\forall, \exists\}$ we observe $\mathcal{Q} = \mathcal{Q}'$. In other words, ψ does not contain any quantifier alternations at all. Therefore, we can shift all quantifiers outwards — existential quantifiers first, renaming bound variables as necessary —, and thus obtain an equivalent formula of the form $\exists\bar{\mathrm{u}}\forall\bar{\mathrm{v}}.\, \varphi'(\bar{\mathrm{u}}, \bar{\mathrm{v}}, \bar{\mathrm{z}})$ with quantifier-free φ'. □

As already pointed out, unfolding formulas in the spirit of Lemma 2.0.3 inevitably incurs immense blowups in the worst case. We will derive upper and lower bounds on the increase in length in subsequent sections (cf. Lemma 3.2.5 and Theorems 3.2.7, 3.5.3, 3.9.9, 3.10.8, and 3.12.5). In particular, we will see that the transformation described in Lemma 2.0.4 leads to formulas that are asymptotically n-fold-exponentially longer than the original.

Another interesting point is that Lemma 2.0.4 holds for first-order formulas irrespective of whether they contain function symbols of arbitrary arity. However, the presence of second-order quantifiers would require additional separateness conditions, similar to the ones strict separateness poses towards quantified subformulas.

In spite of the generality of the result, we will mostly concentrate on relational first-order formulas in the rest of Part I.

Novel Decidable Fragments of Relational First-Order Logic

Separateness of variables will be the prime concept that we use in Chapter 3 to extend well-known fragments of first-order logic that are known to possess a decidable satisfiability problem. The first extended fragment in this family is the *separated fragment*[1], which consists of relational first-order sentences $\exists\bar{\mathrm{z}}\forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1 \ldots \forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n.\, \varphi(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n, \bar{\mathrm{z}})$ in which the sets $\bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_n$ and $\bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_n$ are separated. Lemma 2.0.4 in fact shows that every SF sentence is equivalent to some $\exists^*\forall^*$ prenex sentence. The class of all relational $\exists^*\forall^*$-sentences with equality is called the *Bernays–Schönfinkel–Ramsey fragment (BSR)* and is well known to have a decidable satisfiability problem. A more detailed discussion can be found in Chapter 3. Before we fully embark on the endeavour of finding new decidable first-order fragments, we outline a few other applications of separateness.

Skolemization and Weak and Strong Dependences Between Quantified Variables

In general, nested first-order quantification leads to dependences between existentially and universally quantified variables. Consider the first-order sentence $\varphi := \forall xz\exists y.\, P(x) \leftrightarrow \big(Q(x) \leftrightarrow R(y,z)\big)$. Standard Skolemization removes the quantifier $\exists y$ from φ and replaces every occurrence of the variable y with the term $f(x,z)$ for some fresh *Skolem function* f. The result is the equivalent second-order sentence $\varphi_{\mathrm{Sk}} := \exists f.\, \forall xz.\, P(x) \leftrightarrow \big(Q(x) \leftrightarrow R(f(x,z),z)\big)$. The *Skolem term* $f(x,z)$ makes the dependence of y on the variables x and z explicit. Using the laws of Boolean algebra and quantifier shifting, the original φ can be transformed into the equivalent sentence

$$\begin{aligned}\varphi' := {} & \big(\big(\exists x_1.\, P(x_1) \wedge Q(x_1)\big) \rightarrow \forall z_1\exists y_1.\, R(y_1,z_1)\big) \\ & \wedge \big(\big(\exists x_2.\, P(x_2) \wedge \neg Q(x_2)\big) \rightarrow \forall z_2\exists y_2.\, \neg R(y_2,z_2)\big) \\ & \wedge \big(\big(\exists x_3.\, \neg P(x_3) \wedge Q(x_3)\big) \rightarrow \forall z_3\exists y_3.\, \neg R(y_3,z_3)\big) \\ & \wedge \big(\big(\exists x_4.\, \neg P(x_4) \wedge \neg Q(x_4)\big) \rightarrow \forall z_4\exists y_4.\, R(y_4,z_4)\big)\end{aligned}$$

with a lower quantifier rank. Applying Standard Skolemization to the latter formula replaces every occurrence of y_i with the Skolem term $g_i(z_i)$. This time, we have several Skolem functions $g_1, \ldots, g_4$, each of which has arity one instead of arity two. We will investigate this phenomenon on a semantic level in Chapter 4 and Section 7.2. We shall distinguish two kinds of dependences that occur between existentially quantified variables and universally quantified variables. The dependence of y on x in the original formula φ is a *weak dependence*. A formal definition of weak dependences is given in Definition 4.0.1 on page 111. One characteristic of weak dependences is that they may vanish if $\forall\exists$ quantifier alternations are unfolded in the spirit of Lemma 2.0.3. This is what happens when φ is transformed into φ'. In contrast, the dependence of y on z is considered to be *strong*. Since the two variables co-occur in an atom, the quantifier alternation $\forall z\exists y$ cannot be removed by equivalence-preserving transformations. It turns out that an analysis of separateness in φ can predict a-priori that y only weakly depends on x and that four unary Skolem functions could be used instead of a single binary Skolem function. This leads to a non-standard Skolemization technique that is sensitive to the difference between weak and strong dependences. When we apply it to φ, we obtain the equivalent second-order sentence

$$\varphi'_{\mathrm{Sk}} := \exists g_1 \ldots g_4.\, \forall xz.\, \bigvee_{i=1}^{4} P(x) \leftrightarrow \big(Q(x) \leftrightarrow R(g_i(z),z)\big)\ .$$

In Section 7.2 we shall elaborate on this form of dependence-sensitive Skolemization.

[1]The following remark is intended to clarify any possible confusion regarding terminology. Krom [Kro67] defines the notion of *segregated formulas* which, despite the name similarity to the separated fragment, constitutes a classification of formulas completely orthogonal to the approach used in the present thesis. According to Krom, formulas in conjunctive normal form are *segregated* if every clause either contains positive literals alone or exclusively negative ones. Certain classes of segregated formulas in Krom's sense yield decidable fragments, while others form reduction classes for full first-order logic.

In the field of proof complexity it is known that using different forms of Skolemization can have dramatic effects on the length of shortest refutation proofs [BL94, Egl94]. Hence, the proposed form of Skolemization might be an interesting object of study in that context. Since Skolemization also plays an important role in first-order theorem proving, cf. [NW01, BEL01], analyzing weakness of dependences might lead to significant improvements in this field. Additional inspiration might be drawn from dependency analysis techniques that have been successfully applied in QBF solving (see Remark 7.2.1 on page 191 for references).

So far, we have concentrated on the syntactic side of separateness of variables. There is also a semantic side of this property which we shall study in Chapter 4. We have already mentioned a central idea above, the distinction between weak and strong dependences. Yet another way of applying dependence-sensitive Skolemization is given in the sentence

$$\varphi''_{\mathrm{Sk}} := \exists f g_1 \dots g_4 . \Big(\forall xz . P(x) \leftrightarrow \big(Q(x) \leftrightarrow R(f(x,z),z) \big) \Big) \wedge \forall xz . \bigvee_{i=1}^{4} f(x,z) \approx g_i(z) \ .$$

This sentence consists of two components. The first component is the result of Standard Skolemization φ_{Sk} where the binary Skolem function f is introduced. The second component is a formula that restricts the range of $f(x,z)$ under any model in such a way that for any fixed domain element b in the second argument position, the range of $f(x,\mathsf{b})$ is limited to at most four different values, namely $g_1(\mathsf{b}), g_2(\mathsf{b}), g_3(\mathsf{b}), g_4(\mathsf{b})$. Again, this indicates the weakness of the dependence between y and x. The interpretation of Skolem functions under any model $\mathcal{A}$ of φ is strongly related to *satisfying strategies* (more suggestively: winning strategies) in the *model-checking game* associated with the pair $\langle \varphi, \mathcal{A} \rangle$. This link will become obvious in Chapter 4, where we shall elaborate on such strategies and what weakness of dependences means in the context of model-checking games. If we consider sentences in which all existentially quantified variables are separated from all universally quantified variables, all occurring dependences of existential variables on universal variables are weak. As a consequence, the range of any introduced Skolem functions can be restricted to a finite set. Similarly, there exist satisfying strategies that have a finite image. As the sentence φ contains at least one strong dependence, not every model of $\mathcal{A}$ admits such satisfying strategies with finite images. However, we shall see in Section 4.3 how to construct models accompanied by such a satisfying strategy in certain special cases.

Quantifier Elimination

quantifier elimination

Another area in which separateness of variables might be worth investigating is *quantifier elimination*. For certain logical theories $\mathcal{T}$ over a vocabulary Σ, every first-order Σ-formula can be transformed into a $\mathcal{T}$-equivalent quantifier-free Σ-formula. Two theories for which quantifier elimination-based decision procedures are well-known are *Presburger arithmetic* and *linear arithmetic over the rational numbers (LRA)*. For the former a first decision procedure based on quantifier elimination has been devised by Presburger [Pre29], see also [End72], Section 3.2. A quantifier elimination-based decision procedure for the theory of linear rational arithmetic is the so-called *Fourier–Motzkin elimination method*. It has been (re-)discovered and published several times, e.g. by Fourier in 1826 [Fou26], by Dines in 1919 [Din19], and by Motzkin in 1936 [Mot36]. For more modern accounts, see [DE73, Wil86] and [Sch99], Section 12.2. Historical remarks concerning the (re-)discovery of the method can be found in [Wil86], page 693. For *real arithmetic over polynomials* Tarski [Tar57] devised a procedure for quantifier elimination. A more modern method based on *virtual substitution* is due to Loos and Weispfenning [LW93, Wei97], see below, who were inspired by methods due to Cooper [Coo72] and Ferrante and Rackoff [FR75, FR79]. These methods work also in the LRA setting.[2] Comprehensive accounts and recent results regarding quantifier elimination using the virtual substitution method can be found in the survey articles [Stu17, Stu18] and in the dissertations by Košta [Koš16] and Dolzmann [Dol00]. Given a quantifier-free formula $\psi(\bar{\mathrm{x}}, y)$ over

[2] Validity in the theory of the rationals with addition and multiplication is undecidable. This was proven by Robinson [Rob49] via a reduction of the validity problem for the integers with addition and multiplication. Undecidability of the latter, in turn, was established by Church [Chu36c].

the language of LRA, one can extract a so-called *elimination set* E consisting of pairs $\langle \gamma, t \rangle$ of quantifier-free *guards* γ and *testpoints* t such that

$$\mathbb{Q} \models \forall \bar{\mathrm{x}}.\, \big(\exists y.\, \psi(\bar{\mathrm{x}}, y)\big) \longleftrightarrow \bigvee_{\langle \gamma, t \rangle \in E} \gamma(\bar{\mathrm{x}}) \wedge \psi(\bar{\mathrm{x}}, y)\big[y /\!\!/ t\big] \;, \tag{2.1}$$

where $\psi(\bar{\mathrm{x}}, y)\big[y /\!\!/ t\big]$ denotes a quantifier-free formula which results from virtually substituting y with t and whose free variables belong to $\bar{\mathrm{x}}$. Consult Section 7.1 for further details, in particular the discussion preceding Proposition 7.1.3. The virtual substitution operation $\big[y /\!\!/ t\big]$ maps atoms to quantifier-free formulas and behaves like ordinary substitution on compound formulas. It is worth noting that the extraction of elimination sets for y in ψ is solely based on the atoms in ψ that contain y and their respective *polarity*. Let us for the moment regard formulas of the form $\varphi_1 \to \varphi_2$ as abbreviation for $\neg\varphi_1 \vee \varphi_2$ and $\varphi_1 \leftrightarrow \varphi_2$ as abbreviation for $(\neg\varphi_1 \vee \varphi_2) \wedge (\varphi_1 \vee \neg\varphi_2)$. Then, an occurrence of an atom has *positive polarity*, if it lies within the scopes of an even number of negation signs. If the number is odd, the occurrence has *negative polarity*. Also notice that eliminating a universal quantifier $\forall x$ in a formula $\forall x.\, \chi$ amounts to eliminating $\exists x$ in $\neg\exists x.\, \neg\chi$. Typically, elimination of quantifiers proceeds quantifier by quantifier from innermost to outermost. The reason is simply that often a principle like (2.1) is used to define the elimination procedure in an iterative fashion, where a single quantifier in front of a quantifier-free matrix is eliminated. Of course, it might be beneficial to shift quantifiers beforehand.

Example 2.0.5. *Suppose we intend to eliminate the quantifier $\exists u$ in the formula*

$$\varphi := \; \exists u \exists y \forall x \exists z.\; y < u \;\wedge\; u + 2z = 0 \;\wedge\; (z > 0 \;\wedge\; x + y = 0 \;\to\; x > 0) \;,$$

using the above principle directly means that we have to eliminate $\exists z$, $\forall x$, and $\exists y$ first. When we inspect the co-occurrences of variables in the occurring atoms, we find that the sets $\{u, z\}$ and $\{x\}$ are separated. Hence, after swapping $\exists u$ and $\exists y$, we deal with a sentence of the form $\exists y \exists u \forall x \exists z.\, \chi(y, u, x, z)$ with the mentioned separateness pattern. We transform φ into the equivalent formula

$$\begin{aligned} \varphi' := \exists y.\; & \big(\exists u.\, y < u \;\wedge\; \exists z.\, u + 2z = 0 \;\wedge\; z \leq 0\big) \\ & \vee \Big(\big(\exists u.\, y < u \;\wedge\; \exists z.\, u + 2z = 0\big) \;\wedge\; (\forall x.\, x + y \neq 0 \;\vee\; x > 0)\Big) \;. \end{aligned}$$

After swapping quantifiers $\exists u$ and $\exists z$ in the two subformulas $(\exists u. \ldots)$, we have to deal with the subformulas

$$\varphi_1' := \exists u.\, y < u \;\wedge\; u + 2z = 0 \;\wedge\; z \leq 0 \qquad \textit{and} \qquad \varphi_2' := \exists u.\, y < u \;\wedge\; u + 2z = 0$$

when we intend to eliminate two occurrences of the quantifier $\exists u$. For $\exists u$ in φ_1', we obtain the elimination set $E_1 := \big\{\langle \mathtt{true}, -\infty\rangle, \langle \mathtt{true}, y + \varepsilon\rangle, \langle \mathtt{true}, -2z\rangle\big\}$. Coincidentally, we get the same elimination set for $\exists u$ in φ_2'. Regarding the atoms occurring in φ_1', the virtual substitution operator $\big[u /\!\!/ t\big]$ is defined such that

$$\begin{aligned} (y < u)\big[u /\!\!/ -\infty\big] &= \mathtt{false} \;, & (u + 2z = 0)\big[u /\!\!/ -\infty\big] &= \mathtt{false} \;, \\ (y < u)\big[u /\!\!/ y + \varepsilon\big] &= \mathtt{true} \;, & (u + 2z = 0)\big[u /\!\!/ y + \varepsilon\big] &= \mathtt{false} \;, \\ (y < u)\big[u /\!\!/ -2z\big] &= y < -2z \;, & (u + 2z = 0)\big[u /\!\!/ -2z\big] &= -2z + 2z = 0 \;. \end{aligned}$$

Hence, applying the equivalence principle (2.1) yields that φ_1' is $\mathbb{Q}$-equivalent to

$$\begin{aligned} & \bigvee_{\langle \gamma, t \rangle \in E_1} \big(y < u \;\wedge\; u + 2z = 0 \;\wedge\; z \leq 0\big)\big[u /\!\!/ t\big] \\ & = \; \big(\mathtt{false} \;\wedge\; \mathtt{false} \;\wedge\; z \leq 0\big) \\ & \quad \vee \; \big(\mathtt{true} \;\wedge\; \mathtt{false} \;\wedge\; z \leq 0\big) \\ & \quad \vee \; \big(y < -2z \;\wedge\; -2z + 2z = 0 \;\wedge\; z \leq 0\big) \\ & \eqsim \; y < -2z \;\wedge\; z \leq 0 \;. \end{aligned}$$

We set $\varphi_1'' := y < -2z \ \wedge \ z \leq 0$. *Analogously, elimination of* $\exists u$ *in* φ_2' *yields the formula* $\varphi_2'' := y < -2z$. *Put together, we observe that* φ' *is* $\mathbb{Q}$*-equivalent to*

$$\begin{aligned}
&\exists y \exists z \forall x.\, \big(y < -2z \ \wedge \ z \leq 0\big) \vee \big(y < -2z \ \wedge \ (x + y \neq 0 \ \vee \ x > 0)\big) \\
&\models\!\!\models \ \exists y \exists z \forall x.\, y < -2z \ \wedge \ \big(z \leq 0 \ \vee \ x + y \neq 0 \ \vee \ x > 0\big) \\
&\models\!\!\models \ \exists y \exists z \forall x.\, y < -2z \ \wedge \ \big(z > 0 \ \wedge \ x + y = 0 \ \rightarrow \ x > 0\big) \ .
\end{aligned}$$

Let us call the sentence in the last line φ''. *When we compare this result to the original* φ *and take into account that the extraction of an elimination set* E *for* u *in* φ *is solely based on the atoms in* φ *that contain* u *and their respective polarity, we find that* φ'' *can be considered the result of* directly *eliminating* $\exists u$ *from* φ*, after swapping* $\exists u$ *and* $\exists y$*, by means of the elimination set* $E := \big\{\langle \mathtt{true}, -\infty\rangle, \langle \mathtt{true}, y + \varepsilon\rangle, \langle \mathtt{true}, -2z\rangle\big\}$*:*

$$\begin{aligned}
&\exists y. \bigvee_{\langle \gamma, t\rangle \in E} \big(\forall x \exists z.\, y < u \ \wedge \ u + 2z = 0 \ \wedge \ (z > 0 \ \wedge \ x + y = 0 \ \rightarrow \ x > 0)\big)\big[u /\!\!/ t\big] \\
&= \ \exists y. \big(\forall x \exists z.\, \mathtt{false} \ \wedge \ \mathtt{false} \ \wedge \ (z > 0 \ \wedge \ x + y = 0 \ \rightarrow \ x > 0)\big) \\
&\qquad \vee \ \big(\forall x \exists z.\, \mathtt{true} \ \wedge \ \mathtt{false} \ \wedge \ (z > 0 \ \wedge \ x + y = 0 \ \rightarrow \ x > 0)\big) \\
&\qquad \vee \ \big(\forall x \exists z.\, y < -2z \ \wedge \ -2z + 2z = 0 \ \wedge \ (z > 0 \ \wedge \ x + y = 0 \ \rightarrow \ x > 0)\big) \\
&\models\!\!\models \ \exists y \forall x \exists z.\, y < -2z \ \wedge \ (z > 0 \ \wedge \ x + y = 0 \ \rightarrow \ x > 0) \\
&= \ \varphi'' \ .
\end{aligned}$$

This example illustrates that separateness of variables can facilitate quantifier elimination techniques that break with the from-innermost-to-outermost paradigm. The following proposition is a first step in this direction.[3] It can be proved using a dual, slightly refined variant of Lemma 2.0.3.

Proposition 2.0.6. *Consider a quantifier-free formula* $\varphi(\bar{\mathrm{x}}, \bar{\mathrm{y}}, \bar{\mathrm{z}})$ *over the language of linear rational arithmetic, where* $\bar{\mathrm{x}}$ *and* $\bar{\mathrm{y}}$ *are separated in* φ. *Let* E *be an elimination set for* y_1 *in* φ. *Then*

$$\mathbb{Q} \models \ \Big(\exists y_1 \ldots \exists y_p \forall \bar{\mathrm{x}}.\, \varphi\Big) \longleftrightarrow \Big(\bigvee_{\langle \gamma, t\rangle \in E} \exists y_2 \ldots \exists y_n \forall \bar{\mathrm{x}}.\, \gamma \wedge \varphi\big[y_1 /\!\!/ t\big]\Big) \ .$$

Quantifier elimination techniques are not limited to first-order quantifiers. It is a classical result that any second-order quantifier in relational monadic sentences — sentences containing only unary predicate symbols and equality — can be eliminated. This was discovered by Löwenheim [Löw15], Skolem [Sko19], and Behmann [Beh22]. In relational monadic sentences *without* equality every atom contains at most one first-order variable. Hence, in sentences of the latter kind any first-order variable is trivially separated from all other first-order variables. This high degree of separateness is one of the properties that enable the elimination of second-order quantifiers in Behmann's approach, for instance. We have already mentioned the separated fragment above. It is easy to see that this class of sentences contains every relational monadic first-order sentence. In other words, we are dealing with a syntactic generalization of the class of relational monadic first-order sentences without equality. It is a natural question to ask to what extent a second-order variant of the separated fragment admits elimination of second-order quantifiers. An example given by Ackermann in 1935 [Ack35] already shows severe limitations. Nevertheless, we shall identify a previously unknown class of non-monadic sentences in Section 7.3, from which certain second-order quantifiers can be eliminated.

[3] The presented result is due to Thomas Sturm and Christoph Weidenbach, who wrote it up in an unpublished note in 2014.

Chapter 3

Novel First-Order Fragments with a Decidable Satisfiability Problem

In the early twentieth century David Hilbert initiated his famous program striving for a formalization of the foundations of mathematics.[1] At its core lay the *classical decision problem* of first-order logic: Find an algorithm that determines the validity of any given first-order sentence. Following early pioneering work by Löwenheim [Löw15], Skolem [Sko19], and Behmann [Beh22], the late 1920's and the early 1930's saw first successes in the form of partial solutions by Bernays and Schönfinkel [BS28], Ackermann [Ack28], Herbrand [Her30], Ramsey [Ram30], Gödel [Göd32], Kalmár [Kal33], and Schütte [Sch34a]. All of them have identified classes of first-order sentences for which a decision procedure can be formulated. A turning point was reached when Church [Chu36c] and Turing [Tur36] discovered that the validity problem and, equivalently, the satisfiability problem of first-order logic cannot be solved algorithmically in full generality. It became clear that partial solutions are the best we can hope for. From that point on, the classical decision problem has been understood as the problem of classifying first-order logic into fragments with a decidable or undecidable satisfiability problem. This quest has produced a wealth of positive and also negative results, see [Ack54, Sur59, DG79, Lew79, FLTZ93, BGG97, Hus99, FLHT01] for references. The classification in terms of prefix classes has been solved completely and is comprehensively presented in [BGG97].

the *classical decision problem*

In what follows, we review certain classes of first-order sentences that are known to have a decidable satisfiability problem. For convenience, we shall be less precise every now and then and speak of *decidable fragments* or *decidable classes* in such cases. The following list is intended to give an overview over the fragments that are relevant for the present thesis in one way or another. It is — necessarily — incomplete with respect to all the decidable cases of the classical decision problem that have been studied in the literature over the years. The majority of the listed fragments enjoys the *finite model property*, i.e. every satisfiable sentence in such a fragment has a finite model. This is a sufficient condition for decidability of the associated satisfiability problem. If we can derive a computable upper bound regarding the size of smallest models, we speak of a *small model property*.

decidable fragments of first-order logic

The *monadic first-order fragment (MFO)* comprises all relational first-order sentences without equality that contain only unary predicate symbols. When we refer to the *monadic first-order fragment with equality*, we use the abbreviation $\mathrm{MFO}_{\approx}$.

In the landmark paper by Löwenheim [Löw15] not only the well-known Löwenheim–Skolem theorem was formulated and proved, but also the satisfiability problem for $\mathrm{MFO}_{\approx}$ was shown to be decidable. Hence, $\mathrm{MFO}_{\approx}$ is often referred to as the *Löwenheim fragment*. Skolem [Sko19] and Behmann [Beh22] proved decidability for the *monadic second-order fragment* with equality. Several decades later, Löb [Löb67] and Gurevich [Gur69] extended the positive result for MFO to

[1] Brief historical accounts with a focus on the dawn of computability theory can be found, for instance, in [HU79], Section 7.1 (alternatively: Section 8.2.1 in [HMU01]), and in [Coo04], Chapter 1.

monadic first-order sentences with unary function symbols but without equality, the *Löb–Gurevich fragment*. Moreover, as pointed out by Gurevich [Gur76], the famous decidability result for the *monadic theory of infinite binary trees S2S* by Rabin [Rab69] implies that the satisfiability problem for the class of monadic first-order sentences with equality and a single unary function symbol — the *Rabin fragment* — is decidable. This class is known for containing *infinity axioms*, i.e. satisfiable sentences without a finite model. There are also decision procedures for MFO based on resolution [Joy76, FLTZ93, Lei99, FLHT01] and for $\text{MFO}_{\approx}$ based on superposition [BGW93]. Results concerning the computational complexity of monadic first-order fragments have been obtained by Meyer [Mey74], Rackoff [Rac75], Lewis [Lew78, Lew80], Fürer [Für81], Denenberg and Lewis [DL84a], Compton and Henson [CH90], and Grädel [BGG97]. Satisfiability for MFO and $\text{MFO}_{\approx}$ is NExpTime-complete.

The *Bernays–Schönfinkel–Ramsey fragment (BSR)* comprises all relational first-order sentences in prenex normal form with an $\exists^*\forall^*$ quantifier prefix and with equality.

Bernays and Schönfinkel [BS28] showed that satisfiability for the relational $\exists^*\forall^*$ prefix class without equality is decidable. Today, this class is known as the *Bernays–Schönfinkel fragment (BS)*. Following up, Ramsey [Ram30] added equality to this fragment and also obtained a positive decidability result. This extended class is called *Bernays–Schönfinkel–Ramsey fragment (BSR)* and it is known to posses the finite model property, see [BGG97]. It is interesting to note that Ramsey's article is not so much famous for its contribution to the classical decision problem, but rather for laying the foundation for *Ramsey theory*. Computational complexity results regarding the satisfiability problem for BSR have been obtained by Lewis [Lew78, Lew80], Plaisted [Pla84], and Denenberg and Lewis [DL84a]. The problem is complete for NExpTime. Resolution-based decision procedures are described in [Lei93, FLTZ93]. More recent decision procedures aimed at practical applications include [PV08, PdMB10, AW15]. The fragments BS and BSR have been extended (directly or indirectly) with function symbols in various ways, e.g. by Abadi, Rabinovich, and Sagiv [ARS07, ARS10], Nelson, Dougherty, Fisler, and Krishnamurthi [NDFK12], Korovin [Kor13b], and Ge and de Moura [GdM09]. All of these extensions are carefully formulated so that the finite model property is retained.

The *Ackermann fragment (AF)* comprises all relational first-order sentences in prenex normal form with an $\exists^*\forall\exists^*$ quantifier prefix and without equality.

The satisfiability problem of $\exists^*\forall\exists^*$-sentences without equality was shown to be decidable by Ackermann [Ack28]. In his original proof Ackermann derived the finite model property for AF. The proof published later in [Ack54] proceeds via a reduction to the satisfiability problem for MFO. In [DG79] the finite model property of AF with equality is derived. Resolution-based decision procedures have been devised for AF as well [Joy76, FLTZ93, Lei99]. Moreover, a paramodulation-based decision procedure for AF with equality is also known [FS93]. Gurevich [Gur73] and Maslov and Orevkov [MO72] studied Ackermann sentences with arbitrary function symbols (but without equality). Accordingly, this fragment is called the *Gurevich–Maslov–Orevkov fragment*. While Gurevich proved the finite model property for this fragment, Orevkov and Maslov took a proof-theoretic route based on the inverse method. Another extension of AF is the *Shelah fragment*: $\exists^*\forall\exists^*$-sentences with equality and a single unary function symbol [She77]. This class contains *infinity axioms* and, hence, does not possess the finite model property. A more detailed version of Shelah's proof can be found in Section 7.3 in [BGG97]. Results regarding the computational complexity of the satisfiability problem for the Ackermann fragment and its extensions are due to Lewis [Lew78, Lew80], Fürer [Für81], Grädel [Grä90b], and Kolaitis and Vardi [KV90]. The satisfiability problem for AF is ExpTime-complete.

The *Gödel–Kalmár–Schütte fragment (GKS)* comprises all relational first-order sentences in prenex normal form with an $\exists^*\forall\forall\exists^*$ quantifier prefix and without equality.

Gödel [Göd32, Göd33], Kalmár [Kal33], and Schütte [Sch34a, Sch34b] independently showed that the satisfiability problem for GKS is decidable. Gödel and Kalmár established the finite model

property. A probabilistic proof was later given by Gurevich and Shelah [GS83], see also Section 6.2.3 in [BGG97]. Although Gödel claimed that his proof methods could also be applied for GKS sentences with equality, Goldfarb refuted this claim [Gol84]. However, decidable subclasses are known, e.g. the syntactic subfragments described in [GGS84] and in [Wir76], Section 12. A decidable subclass described in semantic terms is mentioned in Section 6.2.3 in [BGG97]. Computational complexity results have been obtained by Lewis [Lew78, Lew80], and Fürer [Für81, Für83]. Satisfiability for GKS is NExpTime-complete.

The *Skolem fragment* comprises the class of relational $\exists^*\forall^*\exists^*$ prenex sentences without equality that satisfy the following properties. Let $\exists z_1 \dots z_k \forall x_1 \dots x_m \exists y_1 \dots y_n . \psi$ be such a sentence with quantifier-free ψ. Every atom A in ψ is required to contain either (a) at least one of the y_j, or (b) at most one of the x_i, or (c) all $x_1, \dots, x_m$. It is easy to see that GKS is a proper subfragment. Decidability of a slightly more restricted variant of the Skolem fragment was shown in [Sko35]. A resolution-based decision procedure of (an extended variant of) the Skolem fragment is given in [Joy76, FLTZ93, FLHT01].

The *two-variable fragment (FO^2)* comprises all relational first-order sentences with equality that are build up using at most two variables, which may be reused in distinct occurrences of quantifiers.

Scott gave a reduction of the satisfiability problem associated with FO^2 to the satisfiability problem for GKS [Sco62]. This reduction works only for sentences without equality. In 1975 Moritmer [Mor75] proved that FO^2 with equality possess the finite model property. The computational complexity of the satisfiability problem for FO^2 has been determined by Grädel, Kolaitis, and Vardi [GKV97]: it is NExpTime-complete. A resolution-based decision procedure for FO^2 can be found in [HS99]; a tableau-based method is described in [ST08]. A superposition-based decision procedures for FO^2 with equality is devised in [dNP01]. A very recent survey of FO^2 and various extensions is [KPHT18].

Maslov's fragment K comprises all relational first-order sentences without equality that satisfy the following properties (we use the definition from [HS99]). Let φ be any relational sentence in negation normal form and let $\psi(u_1, \dots, u_m)$ be any subformula of φ. We assume that $u_1, \dots, u_m$ are exactly the variables occurring freely in ψ and that they are pairwise distinct. The *φ-prefix of* ψ is the sequence $\mathcal{Q}_1 v_1 \dots \mathcal{Q}_m v_m$ of quantifiers in φ (read from left to right) that bind the free variables of ψ, in particular, we have $\{v_1, \dots, v_m\} = \{u_1, \dots, u_m\}$. The *terminal φ-prefix of* ψ is the longest contiguous suffix of $\mathcal{Q}_1 v_1 \dots \mathcal{Q}_m v_m$ starting with a universal quantifier. Put differently, if $\mathcal{Q}_1 v_1 \dots \mathcal{Q}_m v_m$ is of the form $\exists v_1 \dots v_k \forall v_{k+1} \mathcal{Q}_{k+2} v_{k+2} \dots \mathcal{Q}_m v_m$, then the terminal φ-prefix of ψ is $\forall v_{k+1} \mathcal{Q}_{k+2} v_{k+2} \dots \mathcal{Q}_m v_m$. Notice that the terminal prefix may be empty. The sentence φ belongs to Maslov's fragment K if there are $k \geq 0$ universal quantifiers $\forall x_1, \dots, \forall x_k$ in φ that are not interspersed with existential quantifiers such that for every atom A in φ the terminal φ-prefix of A either (a) is at most of length one, or (b) ends with an existential quantifier, or (c) is of the form $\forall x_1 \dots \forall x_k$.

Maslov introduced K in [Mas68] and devised a decision procedure based on Maslov's *inverse method*[2] [Mas64]. Zamov gave a modern account of Maslov's fragment K and Maslov's decision procedure in a self-contained article [Zam87]. Resolution-based decision procedures are presented in [FLTZ93, HS99, FLHT01], see also Chapter 3 in [Hus99]. To the best knowledge of the author of the present thesis, the computational complexity of the satisfiability problem for Maslov's fragment K is yet unknown. Since K syntactically extends MFO, AF, GKS, and the Skolem fragment (all without equality), the problem must be at least NExpTime-hard.

Maslov's fragment K must not be confused with the *Maslov fragment*, the class of relational Krom sentences without equality in prenex normal form with the quantifier prefix $\exists^*\forall^*\exists^*$. The Maslov fragment is also known to have a decidable satisfiability problem, see Section 8.3.3 in [BGG97]

[2]The inverse method has strong connections to resolution, as noted by Kuehner [Kue71], Zamov [Zam87], Lifschitz [Lif89], and Bachmair and Ganzinger (Section 7.5 in [BG01]).

for more details and references. In particular, Aanderaa and Goldfarb derived the finite model property for this class [AG74].

The *fluted fragment (FL)* comprises all relational first-order sentences without equality that satisfy the following properties. Let $x_1, x_2, x_3, \ldots$ be a fixed ordered sequence of pairwise distinct variables. For every nonnegative integer k we define the set $\mathrm{FL}^{(k)}$ inductively as follows. Any atom $P(x_\ell, \ldots, x_k)$ belongs to $\mathrm{FL}^{(k)}$ — notice that $x_\ell, \ldots, x_k$ is asserted to be a gap-free subsequence of $x_1, x_2, x_3, \ldots$. The set $\mathrm{FL}^{(k)}$ is closed under Boolean combinations, i.e. if φ and ψ belong to $\mathrm{FL}^{(k)}$, then so do $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \to \psi$, $\varphi \leftrightarrow \psi$. Given any $\mathrm{FL}^{(k+1)}$ formula $\varphi(x_1, \ldots, x_{k+1})$, then $\forall x_{k+1}.\, \varphi$ and $\exists x_{k+1}.\, \varphi$ belong to $\mathrm{FL}^{(k)}$. The *fluted fragment (FL)* is the set $\mathrm{FL}^{(0)}$, which contains exclusively sentences. Notice that every sentence φ from $\mathrm{FL}^{(k)}$ can be turned into an equivalent $\mathrm{FL}^{(0)}$ sentence $\forall x_1 \ldots x_k.\, \varphi$.

The fluted fragment was introduced by Quine in two steps [Qui69, Qui76]. In an attempt to extrapolate an extension of MFO from Herbrand's treatment of the fragment in [Her30], Quine [Qui69] considered so-called *homogeneous k-adic* sentences, i.e. FL sentences in which all predicate symbols have arity k. Decidability of the associated satisfiability problem was shown via an extension of Herbrand's proof for MFO [Her30]. Later on, namely at the very end of [Qui76], Quine claimed that the same proof would work for full FL. However, in 1980 Noah pointed out [Noa80] that Quine's decision procedure is only applicable to the subfragment of homogeneous FL sentences. Hence, decidability of the full fragment was considered open again. In a series of articles [Pur96b, Pur96a, Pur99, Pur02] Purdy investigated the fluted fragment and argued that it possess the finite model property (also in the presence of equality). However, several flaws have been detected in Purdy's work [PST16]. For instance, the satisfiability problem associated with FL was believed to be in NExpTime [Pur02], until it was proved to be non-elementary in 2016 [PST16]. A decision procedure for FL based on resolution was devised in [SH00].

Herzig [Her90] considered a class of relational first-order sentences that is very similar to the fluted fragment. *Herzig's ordered fragment* consists of all relational first-order sentences without equality in which every atom $P(v_1, \ldots, v_m)$ satisfies the following property. For every i, $1 \leq i \leq m$, the (unique) quantifier $\mathcal{Q}v_i$ binding v_i lies within the scope of any quantifier $\mathcal{Q}'u$ if and only if $\mathcal{Q}'u$ binds one of the v_j with $j < i$, i.e. $u \in \{v_1, \ldots, v_i\}$. Notice that the definition implies that the $v_1, \ldots, v_m$ are pairwise distinct. While atoms in fluted formulas $\varphi \in \mathrm{FL}^{(k)}$ need to contain a contiguous suffix of the variable sequence $x_1, \ldots, x_k$, any atom A in Herzig's ordered formulas must contain a contiguous prefix of the variables bound by the quantifier sequence governing A. Using the techniques from the proof of Lemma 2.0.3 in an iterated fashion, every sentence from Herzig's ordered fragment can be transformed into an equivalent fluted sentence. This observation is also a corollary of the result we shall develop in Section 3.13 (cf. Lemma 3.13.4 and Proposition 3.13.3).

The *guarded fragment (GF)* comprises all relational first-order sentences with equality that satisfy the following properties. An *atomic guard* $\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ is an atom A such that all $u \in \bar{\mathrm{u}} \cup \bar{\mathrm{v}}$ occur in A. We define the *guarded fragment (GF)* inductively: (i) every relational atom is a GF formula (equality is allowed); (ii) every Boolean combination of GF formulas is a GF formula; (iii) for all tuples $\bar{\mathrm{u}}, \bar{\mathrm{v}}$ and any atomic guard $\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ the following formulas belong to GF: $\forall \bar{\mathrm{u}}.\, \big(\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}}) \to \psi(\bar{\mathrm{u}}, \bar{\mathrm{v}})\big)$ — abbreviated by $\big(\forall \bar{\mathrm{u}}.\, \gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})\big)\psi(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ — and $\exists \bar{\mathrm{u}}.\, \big(\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}}) \wedge \psi(\bar{\mathrm{u}}, \bar{\mathrm{v}})\big)$ — abbreviated by $\big(\exists \bar{\mathrm{u}}.\, \gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})\big)\psi(\bar{\mathrm{u}}, \bar{\mathrm{v}})$. Notice that we assume in any GF formula $\big(\mathcal{Q}\bar{\mathrm{u}}.\, \gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})\big)\psi(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ that all variables that occur freely in ψ also occur in γ.

The guarded fragment was introduced by Andréka, Németi, and van Benthem [ANvB98] as one characterization of the fragment of first-order logic in which propositional modal logic can be embedded via the so-called *standard translation* (cf. Section 2.4 in [BdRV02]). Van Benthem [vB97] also proposed a more liberal form of guards, *loose guards*. A *loose guard* $\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ is a nonempty conjunction of atoms $\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}}) := A_1(\bar{\mathrm{u}}, \bar{\mathrm{v}}) \wedge \ldots \wedge A_k(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ such that all u, v with $u \in \bar{\mathrm{u}}$ and $v \in \bar{\mathrm{u}} \cup \bar{\mathrm{v}}$ co-occur in at least one A_j. The *loosely guarded fragment (LGF)* is then defined by liberalizing (iii) such that loose guards are used instead of atomic guards. In particular, we assume in any LGF formula $\big(\mathcal{Q}\bar{\mathrm{u}}.\, \gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})\big)\psi(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ that (a) all variables that occur freely in ψ also occur

in γ and (b) every variable that is bound by $\mathcal{Q}\bar{\mathrm{u}}$ co-occurs with every free variable from ψ in some atom in γ. Grädel [Grä99b] derived the tree-like model property for GF and LGF and the finite model property for GF. Moreover, the computational complexity of the associated satisfiability problems is pinpointed in the same article: both are complete for deterministic doubly exponential time. A superposition-based decision procedure for GF with equality is due to Ganzinger and de Nivelle [GdN99]. Resolution-based decision procedures for GF and LGF without equality are described in [dN98, GHS00, FLHT01, dNdR03, GHS03]. More variants of guards and guarded quantification have been proposed, which lead to the definition of the *clique-guarded fragment* [Grä99a] and the *packed guarded fragment* [Mar01], for instance. Hodkinson [Hod02] showed that also the loosely guarded fragment, the clique-guarded fragment, and the packed guarded fragment enjoy the finite model property.

Recently, Bárány, ten Cate, and Segoufin [BtCS11, BtCS15] have discovered that guards can be shifted from quantification to negation, see also [Seg17]. This leads to the *guarded-negation first-order fragment (GNFO)*. GNFO comprises all relational first-order formula with equality over the Boolean connectives $\neg, \wedge, \vee$ and existential quantification. Every occurrence of the negation sign is accompanied by a guard, i.e. negation may only occur in the form $\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}}) \wedge \neg\varphi(\bar{\mathrm{v}})$, where γ is an atomic guard and φ is a GNFO formula. In terms of expressiveness, GNFO subsumes GF [BtCS15]. Moreover, in the same article it is shown that GNFO enjoys the tree-like model property and the finite model property. The associated satisfiability problem is complete for deterministic doubly exponential time. Clique-guarded variants of GNFO have also been studied [BtCS15].

The *monadic shallow linear Horn fragment (MSLH)* comprises all finite universally quantified conjunctions $\forall\bar{\mathrm{x}}.\bigwedge_i C_i(\bar{\mathrm{x}})$ of first-order Horn clauses $C_i(\bar{\mathrm{x}})$ without equality satisfying the following properties. Every C_i is of the form $\neg P_1(s_1) \vee \ldots \vee \neg P_n(s_n) \vee Q(t)$ where $n \geq 0$ and the s_k and t are terms. The term t has to be *shallow* and *linear*, i.e. t is either a variable, a constant symbol, or of the form $f(x_1, \ldots, x_m)$ with $m \geq 1$ and pairwise distinct first-order variables $x_1, \ldots, x_m$.

The MSLH clause fragment was introduced by Weidenbach [Wei99], motivated by applications in security, namely, the verification of key-exchange protocols. Driven by applications in program analysis, Nielson, Nielson, and Seidl independently [NNS02] identified a clause fragment, called H_1, that has essentially the same expressive power. This was observed by Goubault-Larrecq [Gou05]. More precisely, Goubault-Larrecq showed that every H_1 sentence φ can be transformed, in polynomial time, into an MSLH sentence ψ such that (i) φ and ψ are equisatisfiable, (ii) $\psi \models \varphi$, and (iii) if both sentences are satisfiable, then their least Herbrand models coincide on the interpretation of every predicate symbol occurring in φ. The analysis of the computational complexity of H_1's satisfiability problem in [NNS02] was sharpened in [Gou05]: it is ExpTime-complete. The MSLH fragment was extended by Teucke and Weidenbach [TW17] to a non-Horn variant. The clause set H_1 was extended in various ways by Seidl and Reuß [SR11, SR12]. Other decidable first-order clause fragments that, like MSLH and H_1, have a very strong connection to tree automata, are described in [Nie96, JMW98, JRV06, SV06, SV08]. Some of them have found applications in the verification of security protocols, see [SV06, SV08] and the references therein.

Other decidable fragments. There are several works that consider undecidable prefix classes, where additional restrictions on the co-occurrences of quantified variables in atoms lead to a decidable fragment [Dre62, DKW62, Gol63, Lew80], see also Section 5.1 in [DG79]. One example is Lewis' fragment T ([Lew80], Section 2A), which comprises first-order sentences of the form $\exists z_1 \ldots z_k \forall x \exists y_1 \ldots y_m \forall x'.\, \psi$ without equality over a relational vocabulary with predicate symbols of arity two only. Moreover, ψ is quantifier-free and may not contain any atoms of the form $P(x', x)$ or $P(x', y_j)$, $1 \leq j \leq m$; atoms of the form $P(x, x')$, $P(x, y_j)$, or $P(y_j, x')$ are allowed, though. This fragment contains infinity axioms, as witnessed by the sentence $\forall x \exists y \forall z.\, P(y, x) \wedge \big(P(x, z) \rightarrow P(y, z)\big) \wedge \neg P(x, x)$ provided in [Lew80]. Nevertheless, the class has a decidable satisfiability problem.

There is a wealth of works focusing on finite clause sets, i.e. $\forall^*$ prenex sentences in conjunctive

normal form after exhaustive Skolemization, rather than sentences. Usually, restrictions are imposed on the syntactic form of clauses, as in the case of the Maslov fragment (relational $\exists^*\forall^*\exists^*$ Krom sentences without equality) or the *Herbrand fragment* (first-order sentences in CNF without equality in which every clause contains exactly one literal). More on decidable Krom clause classes can be found in Section 8.3 in [BGG97]. The satisfiability problem for Herbrand sentences is treated in Section 8.2.2 in [BGG97]; a resolution-based decision procedure was devised by Joyner in [Joy76]. Decidability of the class of Herbrand sentences with equality but only a single unary function symbol was proven by Wirsing in [Wir76], Sections 6–11, via a reduction to the Rabin fragment. Wirsing has also shown that satisfiability for the Herbrand fragment with equality is undecidable [Wir76, Wir77, Wir78].

In later contributions, often a more flexible use of function symbols in clauses is allowed. For known results and references consult the books [FLTZ93, Lei97], the PhD thesis of Hustadt [Hus99], the book chapters [Lei99, FLHT01], and, for instance, the articles [GHS02, LW17]. Exemplary criteria in this direction of research are bounds on the depth at which variables may occur in terms, as witnessed by the *positive variable dominated clause fragment (PVD)*, treated in [FLTZ93, FLHT01]. The generalization of PVD presented in [LW13, LW17] illustrates that there is still room for improvement, and that such improvements may be inspired by real-world applications.

More recently discovered decidable fragments, again based on syntactical restrictions of sentences rather than clause sets, are the *unary negation fragment* [StC13], the *uniform one-dimensional fragment* [KK14], and a family of fragments defined and proved decidable in [MP15, BM17]. Although the unary negation fragment is not syntactically extended by GNFO, the former is subsumed by GNFO in terms of expressiveness [BtCS15]. A close relative of the uniform one-dimensional fragment is the *one-free fragment* introduced in [Tam91, Tam95], see also [FLTZ93, FLHT01].

In the field of knowledge representation, Horn clause sets and equivalent formalisms play a key role. Over the last decade, so-called *existential rules* have attracted a lot of attention, see, e.g., [GHK$^+$13]. Roughly speaking, an existential rule is a first-order implication $\varphi \to \psi$ where φ and ψ are conjunctions of relational atoms and ψ may contain existentially quantified variables. Viewed as a fragment of first-order logic, existential rules in their general form lead to an undecidable satisfiability problem. However, a number of expressive decidable fragments has been discovered. Recent results and further references can be found in [BLM10, CGP10a, CGP10b, BMRT11, KR11, Mug11, LMTV12, GHK$^+$13, BGMR14, BBMR15, ALM17] for example.

In the subsequent sections, we will define novel fragments of first-order logic that all extend some of the above fragments, in particular MFO, BSR, AF, GKS, FO^2, FL, GF, LGF, and GNFO. In essence, the definitions of the new fragments are careful combinations of the concepts of the original definitions with the concept of (strict) separateness of quantified variables. All new fragments still have a decidable satisfiability problem. We will see two approaches to showing decidability of the fragments: an indirect, syntactic approach and a direct, model-theoretic approach. The easier approach shall be presented first for all the fragments. It amounts to devising equivalence-preserving translations — in the spirit of the unfolding techniques presented in Chapter 2 — into fragments that are already known to be decidable. The second approach sheds more light on the underlying semantic properties, in particular on the question which dependences of existentially quantified variables on universally quantified variables are weak and which are strong in the sense already mentioned in Chapter 2. We will present this approach for three of the new fragments in Section 4, namely for the separated fragment, the *generalized Bernays–Schönfinkel–Ramsey fragment*, and the *generalized Ackermann fragment*.

3.1 The Separated Fragment (SF)

We start our exhibition of novel first-order fragments with the simplest, namely the one that we have already briefly introduced in Chapter 2: the *separated fragment*, *SF* for short. Technically, it is defined as a class of prenex sentences, but this is not an essential property. The defining

principle of SF sentences is simply that co-occurrences of universally and existentially quantified variables in atoms are forbidden. Existential variables quantified by *leading* existential quantifiers are exempt from this rule. We consider an existential quantifier *leading* if it does not lie within the scope of any universal quantifier.

Definition 3.1.1 (Separated fragment (SF))**.** *The* separated fragment (SF) *consists of all relational first-order sentences φ with equality that are of the form*

$$\exists \bar{z}\, \forall \bar{x}_1 \exists \bar{y}_1 \ldots \forall \bar{x}_n \exists \bar{y}_n .\, \psi$$

in which ψ is quantifier-free, and in which the sets $\bar{x} := \bar{x}_1 \cup \ldots \cup \bar{x}_n$ and $\bar{y} := \bar{y}_1 \cup \ldots \cup \bar{y}_n$ are separated. The tuples $\bar{z}$ and $\bar{y}_n$ may be empty, i.e. the quantifier prefix does not have to start with an existential quantifier and it does not have to end with an existential quantifier either.

Recall that $\bar{x}$ and $\bar{y}$ are separated in φ if and only if for every atom A occurring in φ we either have $\mathrm{vars}(A) \cap \bar{x} = \emptyset$ or $\mathrm{vars}(A) \cap \bar{y} = \emptyset$. Moreover, notice that the variables in $\bar{z}$ are not subject to any restriction regarding their occurrences.

As already mentioned earlier, Lemma 2.0.4 entails that every SF sentence can be transformed into an equivalent BSR sentence. In order to do so, we just have to replace the subformula $\forall \bar{x}_1 \exists \bar{y}_1 \ldots \forall \bar{x}_n \exists \bar{y}_n .\, \psi$ in an SF sentence φ with an equivalent formula of the form $\exists \bar{u} \forall \bar{v} .\, \psi'$.

Theorem 3.1.2. *Every SF sentence is equivalent to some BSR sentence.*

Since BSR enjoys the finite model property, even if constant symbols are allowed in the syntax, the separated fragment immediately inherits this property. Hence, we conclude that the satisfiability problem for SF (*SF-Sat*) is decidable, even if constant symbols are allowed to occur in SF sentences. SF-Sat

Corollary 3.1.3. *SF enjoys the finite model property and, hence, satisfiability of sentences in SF is decidable. This also holds in the presence of constant symbols in SF sentences.*

It is not hard to see that SF is a proper syntactic extension of BSR. Clearly, the quantified variables in every BSR sentence $\varphi := \exists \bar{z} \forall \bar{x} .\, \psi$ with quantifier-free ψ trivially satisfy the separateness conditions imposed by Definition 3.1.1, as no existential quantifier lies within the scope of any universal quantifier. Similarly, every monadic sentence without equality in prenex normal form trivially satisfies the conditions of Definition 3.1.1, because any monadic atom contains at most one first-order variable. Since any MFO sentence can easily be transformed into an equivalent sentence in prenex normal form, it is fair to say that SF also contains MFO. At the expense of a slightly more technical definition, we could easily modify the definition of the separated fragment so that it is not restricted to prenex sentences. Then, it would indeed contain MFO also in a technically strict sense. On the other hand, it is an easy task to find sentences that belong to SF but neither to BSR nor to MFO. In fact, an interesting example is the sentence φ_2 from Example 2.0.2.

Proposition 3.1.4. *SF properly contains BSR and MFO.*

Another interesting question is whether $\mathrm{MFO}_{\approx}$ is subsumed by SF. Obviously, the sentence $\forall x \exists y .\, x \approx y$ is in $\mathrm{MFO}_{\approx}$ but violates the separateness conditions of SF. Therefore, from the syntactic point of view, there are monadic first-order sentences with equality whose variables are not sufficiently separated for SF. However, the sentence $\forall x \exists y .\, x \approx y$ is equivalent to $\forall x .\, x \approx x$, which certainly belongs to SF and even to BSR. Similarly, we have the $\mathrm{MFO}_{\approx}$ sentence $\forall x \exists y .\, x \not\approx y$, which is not in SF but equivalent to the BSR sentence $\exists y_1 y_2 .\, y_1 \not\approx y_2$. The following theorem witnesses that this is by no means a coincidence. As one consequence, speaking in terms of expressiveness, $\mathrm{MFO}_{\approx}$ is subsumed by BSR and, hence, also by SF.

Theorem 3.1.5. *For every* $\mathrm{MFO}_{\approx}$ *sentence there is an equivalent BSR sentence.*

The proof of this result is based on techniques described by Behmann [Beh22] in the context of second-order quantifier elimination for the monadic second-order fragment. A modern account of these techniques is given in [Wer15a], Section 13.2. For the sake of simplicity, we consider exclusively relational formulas here. However, all arguments can be reused in cases where, in addition, constant symbols are allowed.

Proof sketch. For any positive integer ℓ and any formula $\chi(u)$ we use the abbreviation $\exists^{\geq \ell} u.\, \chi(u)$ to denote the formula $\exists u_1 \dots u_\ell.\, \bigwedge_{i<j} u_i \not\approx u_j \wedge \bigwedge_i \chi(u_i)$, where the variables $u_1, \dots, u_\ell$ are pairwise distinct and do not occur in $\chi(u)$. Moreover, we use the abbreviation $\exists^{< \ell} u.\, \chi(u)$ to denote the formula $\forall u_1 \dots u_\ell.\, \bigvee_{i<j} u_i \approx u_j \vee \bigvee_{1 \leq i \leq \ell} \neg\chi(u_i)$, where the pairwise distinct $u_1, \dots, u_\ell$ do not occur in $\chi(u)$.

We start with two auxiliary results that are dual to one another:

Claim I: Let $\chi(u)$ be a quantifier-free relational monadic formula in which u is the only variable. Let V be a nonempty set of variables, all distinct from u.

(a) The formula $\exists u.\, \chi(u) \wedge \bigwedge_{v \in V} u \not\approx v$ is equivalent to

$$\psi := \Big(\exists^{\geq |V|+1} u.\, \chi(u)\Big) \vee \bigvee_{1 \leq k \leq |V|} \Big(\big(\exists^{\geq k} u.\, \chi(u)\big) \wedge \bigwedge_{\substack{V' \subseteq V \\ |V'|=k}} \Big(\bigvee_{v \in V'} \neg\chi(v) \vee \bigvee_{\substack{v,v' \in V' \\ v \neq v'}} v \approx v' \Big) \Big) .$$

(b) The formula $\forall u.\, \chi(u) \vee \bigvee_{v \in V} u \approx v$ is equivalent to

$$\psi := \Big(\exists^{< |V|+1} u.\, \neg\chi(u)\Big) \wedge \bigwedge_{1 \leq k \leq |V|} \Big(\big(\exists^{< k} u.\, \neg\chi(u)\big) \vee \bigvee_{\substack{V' \subseteq V \\ |V'|=k}} \Big(\bigwedge_{v \in V'} \neg\chi(v) \wedge \bigwedge_{\substack{v,v' \in V' \\ v \neq v'}} v \not\approx v' \Big) \Big) .$$

In both (a) and (b) we observe that every atom A in ψ either exclusively contains free variables or all the variables in A are bound in ψ. ◊

Consider any formula of the form $\psi := \exists y.\, \bigwedge_{i_1} L_{i_1}(y) \wedge \bigwedge_{i_2} y \approx z_{i_2} \wedge \bigwedge_{i_3} y \not\approx z_{i_3}$, where the L_{i_1} are literals over unary predicate symbols and the z_{i_2} and z_{i_3} are all pairwise distinct and different from y. If the subformula $\bigwedge_{i_2} y \approx z_{i_2}$ is nonempty, and thus contains at least one equation $y \approx z$, then the quantifier $\exists y$ can be eliminated and the whole formula ψ is equivalent to $\bigwedge_{i_1} L_{i_1}(z) \wedge \bigwedge_{i_2} z \approx z_{i_2} \wedge \bigwedge_{i_3} z \not\approx z_{i_3}$. Otherwise, Claim I(a) suggests that we can transform ψ into an equivalent formula in which no atom contains a free variable and a bound variable at the same time.

A dual observation holds for any formula $\psi' := \forall x.\, \bigvee_{j_1} K_{j_1}(x) \vee \bigvee_{j_2} x \approx z_{j_2} \vee \bigvee_{j_3} x \not\approx z_{j_3}$. If ψ' contains at least one disequation $x \not\approx z$, then ψ' is equivalent to $\bigvee_{j_1} K_{j_1}(z) \vee \bigvee_{j_2} z \approx z_{j_2} \vee \bigvee_{j_3} z \not\approx z_{j_3}$. Otherwise, Claim I(b) entails that we can transform ψ' into an equivalent formula in which no atom contains a free variable and a bound variable at the same time.

Let $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \dots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n.\, \psi$ be an $\mathrm{MFO}_\approx$ sentence in standard form with quantifier-free ψ. In all the transformations described below, we tacitly assume that formulas are simplified so that they contain neither trivially tautologous nor trivially unsatisfiable subformulas. First, we transform the matrix ψ into a disjunction $\bigvee_k \eta_k(\bar{\mathrm{x}}_1, \dots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1, \dots, \bar{\mathrm{y}}_{n-1}) \wedge \chi_k(\bar{\mathrm{x}}_1, \dots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1, \dots, \bar{\mathrm{y}}_n)$ in which all η_k and χ_k are conjunctions of literals and where each atom in every χ_k contains at least one variable from $\bar{\mathrm{y}}_n$. Hence, φ is equivalent to $\forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \dots \forall \bar{\mathrm{x}}_n.\, \bigvee_k \eta_k \wedge \exists \bar{\mathrm{y}}_n.\, \chi_k$. As we have described above, proceeding variable by variable from $\bar{\mathrm{y}}_n$, every subformula $\exists \bar{\mathrm{y}}_n.\, \chi_k$ can be successively transformed into an equivalent formula $\chi'_k(\bar{\mathrm{x}}_1, \dots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1, \dots, \bar{\mathrm{y}}_{n-1})$ in which no atom contains a free variable and a bound variable at the same time. Therefore, φ is equivalent to $\forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \dots \forall \bar{\mathrm{x}}_n.\, \bigvee_k \eta_k(\bar{\mathrm{x}}_1, \dots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1, \dots, \bar{\mathrm{y}}_{n-1}) \wedge \chi'_k(\bar{\mathrm{x}}_1, \dots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1, \dots, \bar{\mathrm{y}}_{n-1})$. In what follows we treat every subformula $\exists y.\, (\dots)$ in the χ'_k as indivisible unit.

Next, we transform $\bigvee_k \eta_k(\bar{\mathrm{x}}_1, \dots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1, \dots, \bar{\mathrm{y}}_{n-1}) \wedge \chi'_k(\bar{\mathrm{x}}_1, \dots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1, \dots, \bar{\mathrm{y}}_{n-1})$ into a conjunction of disjunctions and treat the quantifier block $\forall \bar{\mathrm{x}}_n$ analogously to how we have treated $\exists \bar{\mathrm{y}}_n$. Notice that afterwards there is no quantifier $\exists y$ with $y \in \bar{\mathrm{y}}_n$ that lies within the scope of any $\forall x$ with $x \in \bar{\mathrm{x}}_n$. Proceeding this way with all the quantifier blocks one after another, we eventually obtain a formula φ' in which, after having been transformed into negation normal form, for every atom A exactly one of three cases applies: (a) A contains only one variable, (b) A contains two existentially quantified variables, or (c) A contains two universally quantified variables. Moreover, there is no nesting of universal and existential quantifiers, i.e. no subformulas

of the form $(\forall x. \ldots (\exists y. \ldots) \ldots)$ or $(\exists y. \ldots (\forall x. \ldots) \ldots)$ occur. Consequently, by shifting quantifiers back outwards again — existential ones first —, we can transform the final result φ' into a BSR sentence. □

The unrestricted presence of function symbols in SF would lead to an undecidable satisfiability problem. Nevertheless, SF could easily be extended so far that it also subsumes the Löb-Gurevich fragment while retaining decidability. We elaborate on this in Section 3.14.

We emphasized in Chapter 2 that transformations based on Lemma 2.0.3, like the one from SF sentences to equivalent BSR sentences, possibly lead to large blowups regarding the length of the formulas. In general, the length of formulas has a significant effect on the size of smallest models. For the BSR case this relation is linear: every satisfiable BSR sentence $\exists y_1 \ldots y_m \forall \bar{x}. \psi$ with $m > 0$ has a model with at most m domain elements.

Proposition 3.1.6 (cf. Proposition 6.2.17 in [BGG97]). *Let $\varphi := \exists \bar{z} \forall \bar{x}. \psi$ be a satisfiable BSR sentence with quantifier-free ψ, possibly containing constant symbols. There is a model $\mathcal{A} \models \varphi$ such that $|\mathsf{A}| \leq \max\big(|\bar{z}| + |\mathrm{consts}(\varphi)|, 1\big)$.*

It turns out that for satisfiable SF sentences φ the size of smallest models in terms of the length of φ cannot be bounded by any tower of exponents $2^{\cdot^{\cdot^{\cdot^{2^{\mathrm{len}(\varphi)}}}}}$ of a fixed height — this is a consequence of Proposition 5.0.1 and Theorem 5.0.3, which we shall prove in Chapter 5. In other words, the asymptotic growth of the size of smallest models is non-elementary in the length of the regarded SF sentence.

Using Lemma 2.0.4 to prove Theorem 3.1.2 does not yield very accurate bounds on the increase in formula length that we incur when translating SF sentences into BSR sentences. We shall conduct a more detailed analysis in the following section, where we derive matching upper and lower bounds. In particular, the entailed upper bound on the size of smallest models for satisfiable SF sentences will have immediate implications with respect to the computational complexity of SF-Sat. The latter will be the subject of Section 5, where we shall also derive corresponding lower bounds and prove that SF-Sat is indeed k-NExpTime-hard for every positive k.

3.2 Translation of SF into BSR: Upper and Lower Bounds

One of the key learning points from the previous sections is Theorem 3.1.2, which says that every SF sentence is equivalent to some BSR sentence. We presented a constructive proof by outlining a procedure that translates any SF sentence given as input into an equivalent BSR sentence. In the present section, we analyze the translation process in more detail. Our goal is to derive upper and lower bounds regarding the length of the resulting BSR sentences. Traditionally, such bounds are formulated in terms of syntactic parameters such as the length of the original SF sentence, the number of predicates it contains, or the number of occurring quantifier alternations. In the case of SF, it turns out that, if we intend to derive accurate bounds that also explain the blowup for subfragments such as MFO, we better take separateness of variables into account. This time, however, we are not interested in existentially and universally quantified variables being separated. In the current context it would not make much sense to define a numerical measure for this kind of separateness, as any sentence which does not exhibit *full* separateness between these two kinds of variables does not belong to SF. What we can measure numerically, though, is the degree of separateness among existentially quantified variables that stem from distinct quantifier blocks.

For convenience, we define this measure in an inverse-proportional way: zero marks the highest possible degree of separateness, larger numbers stand for a lower degree of separateness. Our measure is called the *degree of interaction of existential variables*, denoted by $\partial_\exists(\varphi)$. Intuitively, $\partial_\exists(\varphi)$ an SF sentence φ exhibits a degree $\partial_\exists(\varphi) = k$, if variables from k distinct existential quantifier blocks interact. We say that two variables x, y *interact*, if they co-occur in at least one atom or if there is a third variable z that interacts with both x and y, that is, the property is transitive. For instance, in the SF sentence

$$\varphi := \forall x_1 \exists y_1 v_1 \forall x_2 \exists y_2 v_2 \forall x_3 \exists y_3 v_3. \big(P(x_1, x_2, x_3) \wedge \neg Q(y_1, y_3)\big) \vee P(y_2, v_2, v_3) \vee \neg Q(y_3, v_1)$$

the sets $\{y_1, y_3, v_1\}$ and $\{y_2, v_2, v_3\}$ form the maximal sets of interacting existential variables. Since each of these sets contains variables from at most two distinct quantifier blocks, the formula exhibits a degree $\partial_\exists(\varphi) = 2$.

Definition 3.2.1 (Degree of interaction of existential variables)**.** *Consider any first-order sentence $\varphi := \exists \bar{z}\, \forall \bar{x}_1 \exists \bar{y}_1 \ldots \forall \bar{x}_n \exists \bar{y}_n .\, \psi$ in standard form in which ψ is quantifier free and in which the sets $\bar{x} := \bar{x}_1 \cup \ldots \cup \bar{x}_n$ and $\bar{y} := \bar{y}_1 \cup \ldots \cup \bar{y}_n$ are separated. In addition, we assume that $\bar{x}_1$ and $\bar{y}_1$ are nonempty. The tuple $\bar{z}$ may be empty.*

$\mathrm{idx}(y)$ *For any j, $1 \leq j \leq n$, and any variable $y \in \bar{y}_j$ we say that y* has index *j, denoted* $\mathrm{idx}(y) = j$. *For any nonempty set $Y \subseteq \bar{y}$ of existentially quantified variables and any positive integer k we*

$\partial_\exists(Y, \varphi)$ *say that Y* has degree *k in φ, denoted $\partial_\exists(Y, \varphi) = k$, if k is the maximal number of distinct variables $y_1, \ldots, y_k \in Y$ with $\mathrm{idx}(y_1) < \ldots < \mathrm{idx}(y_k)$. We say that φ*'s degree of interaction of existential variables *(short:* degree*) is k, denoted $\partial_\exists(\varphi) = k$, if k is the smallest positive integer such that we can partition $\bar{y}$ into $m > 0$ parts $Y_1, \ldots, Y_m$ that are all pairwise separated in φ and for which $k = \max\{k_j \mid \partial_\exists(Y_j, \varphi) = k_j, 1 \leq j \leq m\}$. Sentences $\varphi := \exists \bar{z}\, \forall \bar{x}.\, \psi$ in standard form with quantifier-free ψ are said to have* degree *zero, i.e. $\partial_\exists(\varphi) = 0$.*

Any BSR sentence $\exists \bar{z} \forall \bar{x}.\, \psi$ exhibits a degree of zero. There are simply no existentially-quantified variables whose quantifiers lie within the scope of some universal quantifier. Hence, no such variables could interact with any other in atoms. In the context of analyzing the blowup when going from SF to BSR this makes sense. Any reasonable transformation of a BSR sentence into an equivalent BSR sentence does not lead to any blowup.

Consider again the sentences

$$\varphi_1 = \forall x \exists y.\, (P_1(x) \leftrightarrow Q_1(y)) \wedge \ldots \wedge (P_n(x) \leftrightarrow Q_n(y))$$

and

$$\varphi_2 = \forall u \exists v \forall x \exists y.\, (P_1(u, x) \leftrightarrow Q_1(v, y)) \wedge \ldots \wedge (P_n(u, x) \leftrightarrow Q_n(v, y))$$

from Example 2.0.2 (page 16). We observe that $\partial_\exists(\varphi_1) = 1$ and $\partial_\exists(\varphi_2) = 2$. In Example 2.0.2 we showed a transformation of φ_1 into a BSR sentence whose length was singly exponential in the length of φ_1. Similarly, we showed that φ_2 has an equivalent BSR sentence with a length that is doubly exponential in φ_2's length. This already indicates the connection between the degrees of φ_1 and φ_2 and the length of equivalent BSR sentences: a degree of k leads to a k-fold exponential blow up in the worst case. Indeed, we shall derive upper bounds (Lemma 3.2.5) and lower bounds (Theorem 3.2.7) on the blowup that reflect exactly this behavior.

Such fine-grained bounds come in handy when one is also interested in the blowup for subfragments of SF. Recall that MFO is such a subfragment. Obviously, any MFO sentence has a degree of at most one. Hence, the translation of any MFO sentence into BSR incurs an at most singly exponential increase in formula length. In fact, this behavior is not unexpected, as it is well known that any satisfiable MFO sentence φ has a smallest model whose size is at most exponential in the length of φ.

Proposition 3.2.2 (cf. Proposition 6.2.1 in [BGG97])**.** *Let $\varphi := \exists \bar{z}\, \forall \bar{x}_1 \exists \bar{y}_1 \ldots\ \forall \bar{x}_n \exists \bar{y}_n .\, \psi$ be a satisfiable MFO sentence. Moreover, assume that φ contains k distinct predicate symbols. Then, there is a model $\mathcal{A} \models \varphi$ such that $|\mathsf{A}| \leq 2^k$.*

Notice that the shape of the quantifier prefix does not contribute to the upper bound. As we have already pointed out in Proposition 3.1.6, the analogous relationship for satisfiable BSR sentences is linear. Therefore, any translation procedure that maps MFO sentences φ to equivalent BSR sentences ψ having a length doubly exponential in the length of φ, say, must be highly inefficient.

In fact, BSR and MFO belong to the class of SF sentences that have degree at most one. We refer to this subfragment as the *strongly separated fragment (SSF)*.

Definition 3.2.3 (Strongly separated fragment (SSF))**.** *Let $\varphi := \exists \bar{z} \forall \bar{x}_1 \exists \bar{y}_1 \ldots\ \forall \bar{x}_n \exists \bar{y}_n .\, \psi$ be an SF sentence with quantifier-free ψ. We say that φ belongs to the* strongly separated fragment (SSF) *if and only if the sets $\bar{x} := \bar{x}_1 \cup \ldots \cup \bar{x}_n$ and $\bar{y}_1, \ldots, \bar{y}_n$ are all pairwise separated in φ.*

It is easy to find sentences in SSF that are neither in BSR nor in MFO, i.e. SSF is a proper extension of both. Moreover, it is worth noticing that all SF sentences with the quantifier prefix $\exists^*\forall^*\exists^*\forall^*$ belong to the strongly separated fragment. We will see in Chapter 5 that the satisfiability problem for SSF is computationally as hard as satisfiability for BSR and MFO — all three problems are complete for NExpTime.

Next, we conduct the promised analysis of the translation from SF into BSR. Here, we deviate slightly from the presentation in the proof of Lemmas 2.0.4 and 2.0.3. Roughly speaking, in the first phase of the translation process all quantifiers are shifted inwards as far as possible. In order to do so, we first transform the sentence in question into a formula in CNF. After that, we employ the well-known rules of quantifier shifting (cf. Lemma 1.0.1), supplemented with the following lemma.

Lemma 3.2.4. *Let I and K_i, $i \in I$, be sets that are finite, nonempty, and pairwise disjoint. The elements of these sets serve as indices. Let*

$$\varphi := \exists \bar{y}. \bigwedge_{i \in I} \Big(\chi_i(\bar{z}) \vee \bigvee_{k \in K_i} \eta_k(\bar{y}, \bar{z}) \Big)$$

be some first-order formula where the χ_i and the η_k denote arbitrary subformulas that we treat as indivisible units in what follows. We say that any mapping $f : I \to \big(\bigcup_{i \in I} K_i\big)$ is a selection function *if for every $i \in I$ we have $f(i) \in K_i$. We denote the set of all selection functions of this form by $\mathcal{F}$. Then, φ is equivalent to* $\mathcal{F}$

$$\varphi' := \bigwedge_{\substack{S \subseteq I \\ S \neq \emptyset}} \Big(\bigvee_{i \in S} \chi_i(\bar{z}) \Big) \vee \bigvee_{f \in \mathcal{F}} \Big(\exists \bar{y}. \bigwedge_{i \in S} \eta_{f(i)}(\bar{y}, \bar{z}) \Big) \ .$$

Proof. For the sake of readability we sometimes reuse variables in different occurrences of quantifiers in this proof. Using distributivity of $\wedge$ over $\vee$, we transform φ into an equivalent disjunction of conjunctions:

$$\exists \bar{y}. \bigvee_{\substack{\langle T, f \rangle \in \\ (\mathcal{P}I) \times \mathcal{F}}} \Big(\bigwedge_{i \in T} \chi_i(\bar{z}) \Big) \wedge \Big(\bigwedge_{i \in I \setminus T} \eta_{f(i)}(\bar{y}, \bar{z}) \Big) \ .$$

Since the existential quantifier block distributes over the topmost disjunction, we can shift this block inwards and obtain the equivalent formula

$$\bigvee_{\substack{\langle T, f \rangle \in \\ (\mathcal{P}I) \times \mathcal{F}}} \Big(\bigwedge_{i \in T} \chi_i(\bar{z}) \Big) \wedge \Big(\exists \bar{y}. \bigwedge_{i \in I \setminus T} \eta_{f(i)}(\bar{y}, \bar{z}) \Big) \ . \tag{3.1}$$

At this point, we employ distributivity of $\vee$ over $\wedge$ to transform this result into an equivalent conjunction of disjunctions $\varphi'' := \bigwedge_j \bigvee_\ell \psi_{j,\ell}$ in which for every index j and every pair $\langle T, f \rangle \in (\mathcal{P}I) \times \mathcal{F}$ there is *exactly one* ℓ such that either $\psi_{j,\ell} = \chi_i$ for some $i \in T$ or $\psi_{j,\ell} = \exists \bar{y}. \bigwedge_{i \in I \setminus T} \eta_{f(i)}(\bar{y}, \bar{z})$.

In order to show that φ'' is semantically equivalent to φ', we prove the following claims.

Claim I: Every disjunction $\bigvee_\ell \psi_{j,\ell}$ in φ'' is subsumed by a disjunction of the form

$$\psi'_S := \Big(\bigvee_{i \in S} \chi_i(\bar{z}) \Big) \vee \bigvee_{f \in \mathcal{F}} \Big(\exists \bar{y}. \bigwedge_{i \in S} \eta_{f(i)}(\bar{y}, \bar{z}) \Big)$$

for some nonempty $S \subseteq I$.

Proof: Fix some index j and consider $\bigvee_\ell \psi_{j,\ell}$. We set $S := \big\{ i \in I \;\big|\; \psi_{j,\ell} = \chi_i \text{ for some } \ell \big\}$. Consider the set $\overline{S} := I \setminus S$. By definition of S, we know that none of the χ_i with $i \in \overline{S}$ is a constituent of $\bigvee_\ell \psi_{j,\ell}$. For every selection function $f \in \mathcal{F}$ there is some disjunct

$$\Big(\bigwedge_{i \in \overline{S}} \chi_i(\bar{z}) \Big) \wedge \Big(\exists \bar{y}. \bigwedge_{i \in I \setminus \overline{S}} \eta_{f(i)}(\bar{y}, \bar{z}) \Big)$$

in (3.1) of which we know that none of the χ_i in it has been picked as constituent of $\bigvee_\ell \psi_{j,\ell}$ when constructing φ''. Hence, due to the definition of φ'', there must be some ℓ_* such that $\psi_{j,\ell_*} = \exists \bar{\mathrm{y}}.\, \bigwedge_{i \in I \setminus \overline{S}} \eta_{f(i)}(\bar{\mathrm{y}}, \bar{\mathrm{z}})$, where $I \setminus \overline{S} = S$.

Consequently, $\bigvee_\ell \psi_{j,\ell}$ is subsumed by ψ'_S. ◊

Claim II: Each of the subsuming disjunctions ψ'_S in Claim I is indeed equivalent to some disjunction $\bigvee_\ell \psi_{j,\ell}$ in φ''.

Proof: Fix any nonempty $S_* \subseteq I$ and consider ψ'_{S_*}. We obtain the equivalent disjunction ψ_* from the disjuncts in (3.1) as follows. For every $T \subseteq I$ with nonempty $T \cap S_*$ we pick one of the χ_i with $i \in T \cap S_*$ as constituent of ψ_*. For every $T \subseteq I$ for which $T \cap S_*$ is empty and any $f \in \mathcal{F}$ we pick $\exists \bar{\mathrm{y}}.\, \bigwedge_{i \in I \setminus T} \eta_{f(i)}(\bar{\mathrm{y}}, \bar{\mathrm{z}})$ as constituent of ψ_*. Since S_* is nonempty, T must be a proper subset of I and thus $I \setminus T$ is also nonempty.

For every constituent of the form $\exists \bar{\mathrm{y}}.\, \bigwedge_{i \in T'} \eta_{f(i)}(\bar{\mathrm{y}}, \bar{\mathrm{z}})$ that belongs to the disjunction ψ_* we know that $S_* \subseteq T'$. Hence, ψ_* is of the form

$$\Big(\bigvee_{i \in S_*} \chi_i(\bar{\mathrm{z}})\Big) \vee \bigvee_j \bigvee_{f \in \mathcal{F}} \exists \bar{\mathrm{y}}.\, \Big(\bigwedge_{i \in S_*} \eta_{f(i)}(\bar{\mathrm{y}}, \bar{\mathrm{z}})\Big) \wedge \Big(\bigwedge_{i \in S'_j} \eta_{f(i)}(\bar{\mathrm{y}}, \bar{\mathrm{z}})\Big)$$

for certain sets $S'_j \subseteq I \setminus S_*$. Among the S'_j is, in particular, the empty set, originating from $T = I \setminus S_*$. In this case, we have $S'_j = \big(I \setminus T\big) \setminus S_* = \big(I \setminus (I \setminus S_*)\big) \setminus S_* = \emptyset$. Hence, we can equivalently transform ψ_* into

$$\begin{aligned}
&\Big(\bigvee_{i \in S_*} \chi_i(\bar{\mathrm{z}})\Big) \vee \bigvee_{f \in \mathcal{F}} \bigvee_j \exists \bar{\mathrm{y}}.\, \Big(\bigwedge_{i \in S_*} \eta_{f(i)}(\bar{\mathrm{y}}, \bar{\mathrm{z}})\Big) \wedge \Big(\bigwedge_{i \in S'_j} \eta_{f(i)}(\bar{\mathrm{y}}, \bar{\mathrm{z}})\Big) \\
&\models \Big(\bigvee_{i \in S_*} \chi_i(\bar{\mathrm{z}})\Big) \vee \bigvee_{f \in \mathcal{F}} \exists \bar{\mathrm{y}}.\, \bigvee_j \Big(\Big(\bigwedge_{i \in S_*} \eta_{f(i)}(\bar{\mathrm{y}}, \bar{\mathrm{z}})\Big) \wedge \Big(\bigwedge_{i \in S'_j} \eta_{f(i)}(\bar{\mathrm{y}}, \bar{\mathrm{z}})\Big)\Big) \\
&\models \Big(\bigvee_{i \in S_*} \chi_i(\bar{\mathrm{z}})\Big) \vee \bigvee_{f \in \mathcal{F}} \exists \bar{\mathrm{y}}.\, \Big(\bigwedge_{i \in S_*} \eta_{f(i)}(\bar{\mathrm{y}}, \bar{\mathrm{z}})\Big) \vee \bigvee_j \Big(\bigwedge_{i \in S_*} \eta_{f(i)}(\bar{\mathrm{y}}, \bar{\mathrm{z}}) \wedge \bigwedge_{\substack{i \in S'_j \\ S'_j \neq \emptyset}} \eta_{f(i)}(\bar{\mathrm{y}}, \bar{\mathrm{z}})\Big) \\
&\models \Big(\bigvee_{i \in S_*} \chi_i(\bar{\mathrm{z}})\Big) \vee \bigvee_{f \in \mathcal{F}} \exists \bar{\mathrm{y}}.\, \Big(\bigwedge_{i \in S_*} \eta_{f(i)}(\bar{\mathrm{y}}, \bar{\mathrm{z}})\Big) \vee \Big(\Big(\bigwedge_{i \in S_*} \eta_{f(i)}(\bar{\mathrm{y}}, \bar{\mathrm{z}})\Big) \wedge \bigvee_j \bigwedge_{\substack{i \in S'_j \\ S'_j \neq \emptyset}} \eta_{f(i)}(\bar{\mathrm{y}}, \bar{\mathrm{z}})\Big)\,.
\end{aligned}$$

By the absorption axiom of Boolean algebra, we finally obtain the equivalent disjunction

$$\Big(\bigvee_{i \in S_*} \chi_i(\bar{\mathrm{z}})\Big) \vee \bigvee_{f \in \mathcal{F}} \Big(\exists \bar{\mathrm{y}}.\, \bigwedge_{i \in S_*} \eta_{f(i)}(\bar{\mathrm{y}}, \bar{\mathrm{z}})\Big)\,.$$

Thus, the claimed equivalence holds.

We have not yet explicitly argued why the first subformula $\bigvee_{i \in S_*} \chi_i(\bar{\mathrm{z}})$ of ψ_* covers S_* completely. But this is easy to see, when one takes the singleton sets $T = \{i\}$ for every $i \in S_*$ into account, for which we pick the χ_i as a constituent of ψ_*. ◊

This completes the proof of the lemma. □

With Lemma 3.2.4 we now have the right tool at hand to perform the transformations described in the proofs of Lemmas 2.0.4 and 2.0.3 in a way that does not introduce so much redundancy. This will facilitate a neat analysis of the incurred blowup.

Lemma 3.2.5. *Let $\varphi := \exists \bar{\mathrm{z}} \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n.\, \psi$ be an SF sentence of positive degree $\partial_\exists(\varphi)$ in standard form. Let $\mathcal{L}_\varphi(\bar{\mathrm{y}})$ denote the set of all literals in φ that contain at least one variable $y \in \bar{\mathrm{y}} := \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_n$. There exists a sentence $\varphi_{BSR} = \exists \bar{\mathrm{z}}\, \exists \bar{\mathrm{u}}\, \forall \bar{\mathrm{v}}.\, \psi_{BSR}$ in standard form with quantifier-free ψ_{BSR} that is equivalent to φ and contains at most $|\bar{\mathrm{z}}| + |\bar{\mathrm{y}}|^2 \cdot \partial_\exists(\varphi) \cdot \big(2^{\uparrow \partial_\exists(\varphi)}(|\mathcal{L}_\varphi(\bar{\mathrm{y}})|)\big)^{\partial_\exists(\varphi)}$ leading existential quantifiers.*

Proof. For convenience, we pretend that $\bar{z}$ is empty. The argument works for nonempty $\bar{z}$ as well. Let $\bar{x} := \bar{x}_1 \cup \ldots \cup \bar{x}_n$. We transform φ into an equivalent CNF formula of the form

$$\forall \bar{x}_1 \exists \bar{y}_1 \ldots \forall \bar{x}_n \exists \bar{y}_n . \bigwedge_{i \in I} \Big(\chi_i(\bar{x}) \vee \bigvee_{k \in K_i} L_k(\bar{y}) \Big)$$

where I and the K_i are finite, pairwise disjoint sets of indices, the subformulas χ_i are disjunctions of literals, and the L_k are literals. By Lemma 3.2.4, we can construct an equivalent formula of the form

$$\varphi' := \forall \bar{x}_1 \exists \bar{y}_1 \ldots \forall \bar{x}_n . \bigwedge_{S \in \mathcal{P}I \setminus \emptyset} \Big(\bigvee_{i \in S} \chi_i(\bar{x}) \Big) \vee \bigvee_{f \in \mathcal{F}} \Big(\exists \bar{y}_n . \bigwedge_{i \in S} \eta_{f(i)}(\bar{y}) \Big)$$

where $\mathcal{F}$ is the set of all selection functions over the index sets K_i, $i \in I$. For the sake of readability we sometimes reuse variables in different occurrences of quantifiers in this proof. Applying ordinary quantifier shifting, we shift inward the universal quantifier block $\forall \bar{x}_n$ and thus obtain

$$\varphi'' := \forall \bar{x}_1 \exists \bar{y}_1 \ldots \exists \bar{y}_{n-1} . \bigwedge_{S \in \mathcal{P}I \setminus \emptyset} \Big(\forall \bar{x}_n . \bigvee_{i \in S} \chi_i(\bar{x}) \Big) \vee \bigvee_{f \in \mathcal{F}} \Big(\exists \bar{y}_n . \bigwedge_{i \in S} \eta_{f(i)}(\bar{y}) \Big) \, .$$

We now iterate these two steps in an alternating fashion until all quantifier blocks have been shifted inwards in the described way. The constituents of the result $\varphi^{(3)} := \bigwedge_q \Big(\chi_q^{(3)} \vee \bigvee_p \eta_{qp}^{(3)} \Big)$ of this process have the form

$$\chi_q^{(3)} = \forall \bar{x}_1 . \bigvee_{\ell_1} \forall \bar{x}_2 . \bigvee_{\ell_2} \Big(\ldots \Big(\bigvee_{\ell_{n-1}} \forall \bar{x}_n . \bigwedge_{i \in S_{\ell_1, \ldots, \ell_{n-1}}} \chi_i(\bar{x}) \Big) \ldots \Big) \, ,$$

where the $S_{\ell_1, \ldots, \ell_n - 1}$ are certain subsets of I, and

$$\eta_{qp}^{(3)} = \exists \bar{y}_1 . \bigwedge_{\ell'_1} \exists \bar{y}_2 . \bigwedge_{\ell'_2} \Big(\ldots \Big(\bigwedge_{\ell'_{n-1}} \exists \bar{y}_n . \bigwedge_{k \in K_{\ell'_1, \ldots, \ell'_{n-1}}} L_k(\bar{y}) \Big) \ldots \Big) \, ,$$

where the $K_{\ell'_1, \ldots, \ell'_{n-1}}$ are certain subsets of $\bigcup_{i \in I} K_i$.

By definition of $\partial_\exists(\varphi)$, we may assume that there is some positive integer m and a partition of the set $\bar{y}$ into m nonempty subsets $Y_1, \ldots, Y_m$ that are all pairwise separated in φ and such that for every j, $1 \leq j \leq m$, we have $\partial_\exists(Y_j, \varphi) \leq \partial_\exists(\varphi)$. Since the sets $Y_1, \ldots, Y_m$ are pairwise separated in φ, we may partition the set $\mathcal{L}_\varphi(\bar{y})$ into subsets $\mathcal{L}_\varphi(Y_1), \ldots, \mathcal{L}_\varphi(Y_m)$ such that each $\mathcal{L}_\varphi(Y_j)$ contains exactly the literals in φ that contain at least one variable from Y_j. This means, we can rewrite every $\eta_{qp}^{(3)}$ into the form

$$\eta_{qp}^{(4)} = \exists \bar{y}_1 . \bigwedge_{\ell'_1} \exists \bar{y}_2 . \bigwedge_{\ell'_2} \Big(\ldots \Big(\bigwedge_{\ell'_{n-1}} \exists \bar{y}_n . \bigwedge_{j \in [m]} \bigwedge_{k \in K^j_{\ell'_1, \ldots, \ell'_{n-1}}} L_k(Y_j) \Big) \ldots \Big)$$

where the sets $K^1_{\ell'_1, \ldots, \ell'_{n-1}}, \ldots, K^m_{\ell'_1, \ldots, \ell'_{n-1}}$ constitute a partition of $K_{\ell'_1, \ldots, \ell'_{n-1}}$ — some of these parts may be empty. We then observe the following equivalences.

$$\exists \bar{y}_1 . \bigwedge_{\ell'_1} \exists \bar{y}_2 . \bigwedge_{\ell'_2} \Big(\ldots \Big(\bigwedge_{\ell'_{n-1}} \exists \bar{y}_n . \bigwedge_{j \in [m]} \bigwedge_{k \in K^j_{\ell'_1 \ldots \ell'_{n-1}}} L_k(Y_j) \Big) \ldots \Big)$$

$$\models \; \exists \bar{y}_1 . \bigwedge_{\ell'_1} \exists \bar{y}_2 . \bigwedge_{\ell'_2} \Big(\ldots \Big(\bigwedge_{\ell'_{n-1}} \bigwedge_{j \in [m]} \exists (\bar{y}_n \cap Y_j) . \bigwedge_{k \in K^j_{\ell'_1 \ldots \ell'_{n-1}}} L_k(Y_j) \Big) \ldots \Big)$$

$$\models \exists \bar{y}_1 . \bigwedge_{\ell'_1} \exists \bar{y}_2 . \bigwedge_{\ell'_2} \Big(\dots \Big(\bigwedge_{j \in [m]} \bigwedge_{\ell''_{n-1}} \exists (\bar{y}_n \cap Y_j) . \bigwedge_{k \in K^j_{\ell'_1 \dots \ell'_{n-2} \ell''_{n-1}}} L_k(Y_j) \Big) \dots \Big)$$

$$\vdots$$

$$\models \bigwedge_{j \in [m]} \exists (\bar{y}_1 \cap Y_j) . \bigwedge_{\ell''_1} \exists (\bar{y}_2 \cap Y_j) . \bigwedge_{\ell''_2} \Big(\dots \Big(\bigwedge_{\ell''_{n-1}} \exists (\bar{y}_n \cap Y_j) . \bigwedge_{k \in K^j_{\ell''_1 \dots \ell''_{n-1}}} L_k(Y_j) \Big) \dots \Big)$$

For every $\eta^{(4)}_{qp}$ we call the result of the above transformation $\eta^{(5)}_{qp}$. In cases where the set $\bar{y}_i \cap Y_j$ is empty, the existential quantifier block vanishes. For every $j \in [m]$ there are at most $\partial_\exists(Y_j, \varphi)$ nonempty sets $\bar{y}_i \cap Y_j$. Hence, every $\eta^{(5)}_{qp}$ contains at most $\partial_\exists(\varphi)$ nested existential quantifier blocks that are separated by in-between conjunctive connectives in the syntax tree.

We obtain $\varphi^{(5)}$ from $\varphi^{(3)}$ by replacing every constituent $\eta^{(3)}_{qp}$ with the corresponding $\eta^{(5)}_{qp}$ after applying the idempotence axioms of Boolean Algebra exhaustively to remove redundant conjuncts.

Let $\kappa := \max\{|\mathcal{L}_\varphi(Y_j)| \mid 1 \leq j \leq m\}$. Due to the idempotence axioms, the following upper bounds can be shown inductively for any positive integer d, starting from $d = 1$: Modulo idempotence, there are at most $2^{\uparrow d}(\kappa)$ formulas of the form

$$\exists (\bar{y}_{i_1} \cap Y_j) . \bigwedge_{\ell_1} \exists (\bar{y}_{i_2} \cap Y_j) . \bigwedge_{\ell_2} \Big(\dots \Big(\bigwedge_{\ell_{d-1}} \exists (\bar{y}_{i_d} \cap Y_j) . \bigwedge_{k \in K'^j_{\ell_1, \dots, \ell_{d-1}}} L_k(Y_j) \Big) \dots \Big) .$$

For the sentence $\varphi^{(5)} = \bigwedge_q \Big(\chi^{(3)}_q \vee \bigvee_p \eta^{(5)}_{qp} \Big)$ this means that it contains at most $m \cdot 2^{\uparrow \partial_\exists(\varphi)}(\kappa)$ distinct subformulas (not occurrences thereof!) that are of the form $\exists y . \psi'$ and do not lie within the scope of any quantifier. We treat every such subformula $\exists y . \psi'$ and every subformula $\chi^{(3)}_q$ as indivisible unit and, employing distributivity of $\wedge$ over $\vee$, transform $\varphi^{(5)}$ into a disjunction of conjunctions $\varphi^{(6)} := \bigvee_s \Big(\bigwedge_{r_1} \chi^{(6)}_{r_1} \wedge \bigwedge_{r_2} \eta^{(6)}_{r_2} \Big)$ where the $\chi^{(6)}_{r_1}$ have the same shape as the $\chi^{(3)}_q$, and the $\eta^{(6)}_{r_2}$ are of the form

$$\exists (\bar{y}_{i_1} \cap Y_j) . \bigwedge_{\ell_1} \exists (\bar{y}_{i_2} \cap Y_j) . \bigwedge_{\ell_2} \Big(\dots \Big(\bigwedge_{\ell_{d_j - 1}} \exists (\bar{y}_{i_{\partial_j}} \cap Y_j) . \bigwedge_{k \in K'^j_{\ell_1, \dots, \ell_{d_j - 1}}} L_k(Y_j) \Big) \dots \Big)$$

for some j and certain indices $i_1, \dots, i_{d_j}$ with $1 \leq i_1 < \dots < i_{d_j} \leq n$, all depending on r_2; d_j abbreviates the expression $\partial_\exists(Y_j, \varphi)$.

Due to previous observations, we know that, modulo idempotence, r_2 ranges over at most $m \cdot 2^{\uparrow \partial_\exists(\varphi)}(\kappa)$ indices. Moreover, any ℓ_k in any $\eta^{(6)}_{r_2}$ ranges over at most $2^{\uparrow \partial_\exists(\varphi) - k}(\kappa)$ indices. Consequently, every constituent $\bigwedge_{r_2} \eta^{(6)}_{r_2}$ in $\varphi^{(6)}$ contains at most $m \cdot \max_{i,j} |\bar{y}_i \cap Y_j| \cdot \sum_{k'=1}^{\partial_\exists(\varphi)} \prod_{d=k'}^{\partial_\exists(\varphi)} 2^{\uparrow d}(\kappa)$ occurrences of existential quantifiers.

Since these existential quantifiers distribute over the topmost disjunction when we shift them outwards to the front of the sentence $\varphi^{(6)}$, and since the universal quantifiers in the $\chi^{(6)}_q$ may also be shifted back outwards, we have shown that φ is equivalent to some BSR sentence with at most $|\bar{y}|^2 \cdot \partial_\exists(\varphi) \cdot \big(2^{\uparrow \partial_\exists(\varphi)}(\kappa)\big)^{\partial_\exists(\varphi)}$ leading existential quantifiers. □

Put together, Proposition 3.1.6 and Lemma 3.2.5 immediately entail the following small model property for SF.

Theorem 3.2.6 (Small model property for SF)**.** *Every satisfiable SF sentence φ has a model whose domain contains most* $\mathrm{len}(\varphi) + \big(\mathrm{len}(\varphi)\big)^2 \cdot \partial_\exists(\varphi) \cdot \big(2^{\uparrow \partial_\exists(\varphi)}(\mathrm{len}(\varphi))\big)^{\partial_\exists(\varphi)}$ *domain elements.*

In cases where $\partial_\exists(\varphi) = 1$, the bound in Theorem 3.2.6 simplifies to $\mathrm{len}(\varphi) + \big(\mathrm{len}(\varphi)\big)^2 \cdot 2^{\mathrm{len}(\varphi)}$. This leads to a small model property for SSF, and its subfragment MFO, that stipulates for

satisfiable sentences the existence of a model of exponential size in the length of the formula. Concerning the asymptotic growth, this yields a reasonable upper bound on the size of small models of satisfiable MFO sentences that is not too far away from Proposition 3.2.2. This works in spite of the fact that MFO sentences may contain arbitrarily nested alternating quantifiers.

Next, we complement the obtained upper bound on the length of the BSR sentences resulting from the translation of SF sentences with a corresponding non-elementary lower bound.

Theorem 3.2.7. *There is a class of satisfiable SF sentences that are Horn and Krom such that for every positive integer n the class contains a sentence φ of degree $\partial_\exists(\varphi) = n$ and with a length polynomial in n for which any equivalent BSR sentence contains at least $\sum_{k=1}^{n} 2^{\uparrow k}(n)$ leading existential quantifiers.*

Proof. Let $n \geq 1$ be some positive integer. Consider the following first-order sentence in which the sets $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_n\}$ are separated:

$$\varphi := \forall x_n \exists y_n \ldots \forall x_1 \exists y_1 . \bigwedge_{i=1}^{4n} \big(P_i(x_1, \ldots, x_n) \leftrightarrow Q_i(y_1, \ldots, y_n)\big) \; . \qquad \varphi$$

Notice that we change the orientation of the indices in the quantifier prefix in this proof.

In order to construct a particular model of φ, we inductively define the following sets: $\mathcal{S}_1 := \big\{S \subseteq [4n] \;\big|\; |S| = 2n\big\}$, $\mathcal{S}_{k+1} := \big\{S \in \mathcal{P}\mathcal{S}_k \;\big|\; |S| = \frac{1}{2} \cdot |\mathcal{S}_k|\big\}$ for every $k > 1$. Then, we observe that $\mathcal{S}_k$

$$\begin{aligned}
|\mathcal{S}_1| &= \tbinom{4n}{2n} \geq \big(\tfrac{4n}{2n}\big)^{2n} = 2^{2n}, \\
|\mathcal{S}_2| &= \tbinom{|\mathcal{S}_1|}{|\mathcal{S}_1|/2} \geq \big(\tfrac{|\mathcal{S}_1|}{|\mathcal{S}_1|/2}\big)^{|\mathcal{S}_1|/2} = 2^{|\mathcal{S}_1|/2} \geq 2^{2^{2n}/2} = 2^{2^{2n-1}}, \\
&\vdots \\
|\mathcal{S}_n| &= \tbinom{|\mathcal{S}_{n-1}|}{|\mathcal{S}_{n-1}|/2} \geq 2^{|\mathcal{S}_{n-1}|/2} \geq 2^{2^{\vdots^{2^{2n-1}-1}}-1} \geq 2^{\uparrow n}(2n-(n-1)) = 2^{\uparrow n}(n+1),
\end{aligned}$$

where the inequality $\binom{n}{k} \geq (n/k)^k$ can be found in [CSRL01] (page 1097), for example.

Having the sets $\mathcal{S}_k$, we now define the structure $\mathcal{A}$ as follows: $\mathcal{A}$

$$\mathsf{A} := \textstyle\bigcup_{k=1}^{n} \big\{\mathsf{a}_S^{(k)}, \mathsf{b}_S^{(k)} \;\big|\; S \in \mathcal{S}_k\big\},$$

$$P_i^{\mathcal{A}} := \big\{\langle \mathsf{a}_{S_1}^{(1)}, \ldots, \mathsf{a}_{S_n}^{(n)}\rangle \in \mathsf{A}^n \;\big|\; i \in S_1 \in S_2 \in \ldots \in S_n\big\} \text{ for } i = 1, \ldots, 4n, \text{ and}$$

$$Q_i^{\mathcal{A}} := \big\{\langle \mathsf{b}_{S_1}^{(1)}, \ldots, \mathsf{b}_{S_n}^{(n)}\rangle \in \mathsf{A}^n \;\big|\; i \in S_1 \in S_2 \in \ldots \in S_n\big\} \text{ for } i = 1, \ldots, 4n.$$

Clearly, for any choice of $S_1, \ldots, S_n$ and every i, $1 \leq i \leq 4n$, we have

$$\mathcal{A}, \big[x_1 \mapsto \mathsf{a}_{S_1}^{(1)}, \ldots, x_n \mapsto \mathsf{a}_{S_n}^{(n)}, y_1 \mapsto \mathsf{b}_{S_1}^{(1)}, \ldots, y_n \mapsto \mathsf{b}_{S_n}^{(n)}\big] \models P_i(x_1, \ldots, x_n) \leftrightarrow Q_i(y_1, \ldots, y_n) \; .$$

For any other choice of tuples $\langle \mathsf{c}_1, \ldots, \mathsf{c}_n\rangle$, i.e. there do not exist sets $S_1 \in \mathcal{S}_1, \ldots, S_n \in \mathcal{S}_n$ such that $\langle \mathsf{c}_1, \ldots, \mathsf{c}_n\rangle$ equals $\langle \mathsf{a}_{S_1}^{(1)}, \ldots, \mathsf{a}_{S_n}^{(n)}\rangle$ or $\langle \mathsf{b}_{S_1}^{(1)}, \ldots, \mathsf{b}_{S_n}^{(n)}\rangle$, we observe $\mathcal{A}, [x_1 \mapsto \mathsf{c}_1, \ldots, x_n \mapsto \mathsf{c}_n] \not\models P_i(x_1, \ldots, x_n)$ and $\mathcal{A}, [y_1 \mapsto \mathsf{c}_1, \ldots, y_n \mapsto \mathsf{c}_n] \not\models Q_i(y_1, \ldots, y_n)$ for every i. Hence,

$$\mathcal{A}, [x_1 \mapsto \mathsf{c}_1, \ldots, x_n \mapsto \mathsf{c}_n, y_1 \mapsto \mathsf{c}_1, \ldots, y_n \mapsto \mathsf{c}_n] \models \bigwedge_{i=1}^{4n} P_i(x_1, \ldots, x_n) \leftrightarrow Q_i(y_1, \ldots, y_n) \; .$$

Consequently, $\mathcal{A}$ is a model of φ.

Consider the following simple two-player game with Players $\mathfrak{A}$ and $\mathfrak{B}$ where both players have complete and instantaneous knowledge about all moves that are made by either player. In the first round $\mathfrak{A}$ moves first by picking some domain element $\mathsf{a}_{S_{\mathfrak{A},n}}^{(n)}$ for some set $S_{\mathfrak{A},n} \in \mathcal{S}_n$. $\mathfrak{B}$ knows $\mathfrak{A}, \mathfrak{B}$ $S_{\mathfrak{A},j}$, $\mathsf{a}_{S_{\mathfrak{A},j}}^{(j)}$

$S_{\mathfrak{B},j}$, $\mathsf{b}^{(j)}_{S_{\mathfrak{B},j}}$

about $\mathfrak{A}$'s choice and answers by picking a domain element $\mathsf{b}^{(n)}_{S_{\mathfrak{B},n}}$ for some set $S_{\mathfrak{B},n} \in \mathcal{S}_n$. The game continues for $n-1$ more rounds, where in every round Player $\mathfrak{A}$ picks a domain element $\mathsf{a}^{(j)}_{S_{\mathfrak{A},j}}$ with $S_{\mathfrak{A},j} \in S_{\mathfrak{A},j+1}$ and $\mathfrak{B}$ answers by picking some $\mathsf{b}^{(j)}_{S_{\mathfrak{B},j}} \in S_{\mathfrak{B},j+1}$. Hence, in the last round the chosen domain elements $\mathsf{a}^{(1)}_{S_{\mathfrak{A},1}}$ and $\mathsf{b}^{(1)}_{S_{\mathfrak{B},1}}$ are such that $S_{\mathfrak{A},1}$ and $S_{\mathfrak{B},1}$ are both nonempty subsets of $[4n]$. Player $\mathfrak{A}$ wins if and only if

$$\mathcal{A}, \left[x_1 \mapsto \mathsf{a}^{(1)}_{S_{\mathfrak{A},1}}, \ldots, x_n \mapsto \mathsf{a}^{(n)}_{S_{\mathfrak{A},n}}, y_1 \mapsto \mathsf{b}^{(1)}_{S_{\mathfrak{B},1}}, \ldots, y_n \mapsto \mathsf{b}^{(n)}_{S_{\mathfrak{B},n}}\right] \not\models P_i(x_1, \ldots, x_n) \leftrightarrow Q_i(y_1, \ldots, y_n)$$

for some $i \in [4n]$, and Player $\mathfrak{B}$ wins if and only if

$$\mathcal{A}, \left[x_1 \mapsto \mathsf{a}^{(1)}_{S_{\mathfrak{A},1}}, \ldots, x_n \mapsto \mathsf{a}^{(n)}_{S_{\mathfrak{A},n}}, y_1 \mapsto \mathsf{b}^{(1)}_{S_{\mathfrak{B},1}}, \ldots, y_n \mapsto \mathsf{b}^{(n)}_{S_{\mathfrak{B},n}}\right] \models P_i(x_1, \ldots, x_n) \leftrightarrow Q_i(y_1, \ldots, y_n)$$

for every $i \in [4n]$. Since $\mathcal{A}$ is a model of φ, there must exist a winning strategy for $\mathfrak{B}$.

Claim I: There is exactly one winning strategy for $\mathfrak{B}$, namely, for every $j = n, \ldots, 1$ Player $\mathfrak{B}$ picks the element $\mathsf{b}^{(j)}_{S_{\mathfrak{A},j}}$ in round $n - j + 1$, i.e. for every j we have $S_{\mathfrak{B},j} = S_{\mathfrak{A},j}$.

Proof: It is easy to see that the described strategy is a winning strategy for $\mathfrak{B}$.

Assume $\mathfrak{B}$ deviates from this strategy. This means there exists some j_*, $1 \leq j_* \leq n$, such that $\mathfrak{B}$ did not adhere to the described strategy in the $(n - j_* + 1)$st round, i.e. $S_{\mathfrak{B},j_*} \neq S_{\mathfrak{A},j_*}$.

We show by induction on j_* that $\mathfrak{A}$ has a winning strategy from this deviation point on.

For the base case $j_* = 1$ we consider two distinct nonempty sets $S_{\mathfrak{A},1}, S_{\mathfrak{B},1} \subseteq [4n]$. There must be some index i_* that belongs to one of the two sets but not to the other, i.e. $i_* \in (S_{\mathfrak{A},1} \cup S_{\mathfrak{B},1}) \setminus (S_{\mathfrak{A},1} \cap S_{\mathfrak{B},1})$.

Suppose that $i_* \in S_{\mathfrak{A},1} \setminus S_{\mathfrak{B},1}$. Hence, we can construct the chain $i_* \in S_{\mathfrak{A},1} \in \ldots \in S_{\mathfrak{A},n}$, by definition of the allowed moves. This entails $\mathcal{A}, [x_1 \mapsto \mathsf{a}^{(1)}_{S_{\mathfrak{A},1}}, \ldots, x_n \mapsto \mathsf{a}^{(n)}_{S_{\mathfrak{A},n}}] \models P_{i_*}(x_1, \ldots, x_n)$.

On the other hand, we know $\mathcal{A}, [y_1 \mapsto \mathsf{b}^{(1)}_{S_{\mathfrak{B},1}}, \ldots, y_n \mapsto \mathsf{b}^{(n)}_{S_{\mathfrak{B},n}}] \not\models Q_{i_*}(y_1, \ldots, y_n)$, because of $i_* \notin S_{\mathfrak{B},1}$. Hence, $\mathfrak{A}$ wins and the chosen strategy cannot be a winning strategy for $\mathfrak{B}$.

The case where $i_* \in S_{\mathfrak{B},1} \setminus S_{\mathfrak{A},1}$ is symmetric and $\mathfrak{A}$ also wins.

For the inductive case we fix some $j_* > 1$. Since $S_{\mathfrak{A},j_*}$ and $S_{\mathfrak{B},j_*}$ are distinct but have the same number of elements, there is some set $S' \in S_{\mathfrak{A},j_*} \setminus S_{\mathfrak{B},j_*}$. If $\mathfrak{A}$ picks $\mathsf{a}^{(j_*-1)}_{S_{\mathfrak{A},j_*-1}} := \mathsf{a}^{(j_*-1)}_{S'}$ in the following round, we have $S_{\mathfrak{B},j_*-1} \neq S_{\mathfrak{A},j_*-1}$ for any choice $\mathsf{b}^{(j_*-1)}_{S_{\mathfrak{B},j_*-1}}$ that $\mathfrak{B}$ could possibly make. By induction, $\mathfrak{A}$ has a winning strategy starting from the next round of the game. Hence, there is a winning strategy starting from the current round. ◊

The just proved claim would still hold true if we allowed $\mathfrak{B}$ to freely pick any element of the domain A at every round. The reason is that for any choice of elements $\mathsf{a}^{(n)}_{S_{\mathfrak{A},n}}, \ldots, \mathsf{a}^{(1)}_{S_{\mathfrak{A},1}}$ made by $\mathfrak{A}$ with $S_{\mathfrak{A},1} \in \ldots \in S_{\mathfrak{A},n} \in \mathcal{S}_n$ we know that $S_{\mathfrak{A},1}$ is nonempty. Hence, we can always find some $i_* \in S_{\mathfrak{A},1}$ such that $\langle \mathsf{a}^{(n)}_{S_{\mathfrak{A},n}}, \ldots, \mathsf{a}^{(1)}_{S_{\mathfrak{A},1}} \rangle \in P^{\mathcal{A}}_{i_*}$. On the other hand, for any sequence $\mathsf{c}_n, \ldots, \mathsf{c}_1$ picked by $\mathfrak{B}$ that does not comply with the rules of the described game, we have $\langle \mathsf{c}_n, \ldots, \mathsf{c}_1 \rangle \notin Q^{\mathcal{A}}_{i_*}$.

This result proves the following observation.

Claim II: For any of the $\mathsf{b}^{(k)}_S$ the substructure of $\mathcal{A}$ induced by the domain $\mathsf{A} \setminus \{\mathsf{b}^{(k)}_S\}$ does not satisfy φ.

Proof: The reason is simply that in this case player $\mathfrak{A}$ can always prevent $\mathfrak{B}$ from reaching a state of the game where $\mathfrak{B}$ can apply the described winning strategy. ◊

We have already analyzed the size of the sets $\mathcal{S}_k$. Due to the observed lower bounds, we know that A contains at least $\sum_{k=1}^{n} 2^{\uparrow k}(n)$ elements of the form $\mathsf{b}^{(k)}_S$.

Next, we argue that any $\exists^*\forall^*$-sentence φ_* that is semantically equivalent to φ must contain at least $\sum_{k=1}^{n} 2^{\uparrow k}(n)$ leading existential quantifiers. Let $\varphi_* := \exists y_1 \dots y_m \forall x_1 \dots x_\ell .\, \chi_*$ with quantifier-free χ_* be a sentence with minimal m that is semantically equivalent to φ. Since $\mathcal{A}$ is also a model of φ_*, we know that there is a sequence of elements $\mathsf{c}_1, \dots, \mathsf{c}_m$ taken from the domain A such that $\mathcal{A}, [y_1 \mapsto \mathsf{c}_1, \dots, y_m \mapsto \mathsf{c}_m] \models \forall x_1 \dots x_\ell .\, \chi_*$. Consequently, we can extend $\mathcal{A}$ to a model $\mathcal{A}_*$ (over the same domain) of the Skolemized sentence $\varphi_{\mathrm{Sk}} := \forall x_1 \dots x_\ell .\, \chi_*\big[y_1/c_1, \dots, y_m/c_m\big]$ $\mathcal{A}_*$ by adding $c_j^{\mathcal{A}_*} := \mathsf{c}_j$ for $j = 1, \dots, m$. On the other hand, every model of φ_{Sk} is also a model of φ_*. The vocabulary underlying φ_{Sk} comprises exactly the constant symbols $c_1, \dots, c_m$ and does not contain any other function symbols. Suppose $m < \sum_{k=1}^{n} 2^{\uparrow k}(n)$. Hence, there is some $\mathsf{b}_S^{(k)}$ with $S \in \mathcal{S}_k$ such that for every j we have $c_j^{\mathcal{A}_*} \neq \mathsf{b}_S^{(k)}$. By the Substructure Lemma, the following substructure $\mathcal{B}$ of $\mathcal{A}_*$ constitutes a model of φ_{Sk}: $\mathsf{B} := \mathsf{A}_* \setminus \{\mathsf{b}_S^{(k)}\}$, $P_i^{\mathcal{B}} := P_i^{\mathcal{A}_*} \cap \mathsf{B}^n = P_i^{\mathcal{A}_*}$ and $Q_i^{\mathcal{B}} := Q_i^{\mathcal{A}_*} \cap \mathsf{B}^n$ for every i, and $c_j^{\mathcal{B}} := c_j^{\mathcal{A}_*}$ for every j. But then $\mathcal{B}$ must also be a model of both φ_* and φ, since every model of φ_{Sk} is a model of φ_*, and because we assumed φ_* and φ to be equivalent. This contradicts Claim II, and thus we must have $m \geq \sum_{k=1}^{n} 2^{\uparrow k}(n)$.

Since every atom $Q_i(y_1, \dots, y_n)$ contains n variables from existential quantifier blocks that are interspersed with universal quantifier blocks, the degree $\partial_\exists(\varphi)$ of φ is n. Moreover, φ can easily be transformed into a CNF that is Horn and Krom at the same time. Hence, the theorem holds. □

Theorem 3.2.7 entails that there is no elementary upper bound on the length of the BSR sentences that result from any equivalence-preserving transformation of SF sentences into BSR. On the other hand, by Lemma 3.2.5, there is an elementary upper bound, if we only consider SF sentences with a bounded degree of interaction of existential variables. A special case of Theorem 3.2.7 highlights the difference in succinctness between BSR and MFO. By Thereom 3.2.5, we already know that every MFO sentence is equivalent to some BSR sentence whose length is at most exponential in the length of the original MFO sentence. The following proposition entails that, in the worst case, this exponential blowup cannot be avoided.

Proposition 3.2.8. *There is a class of MFO sentences that are Horn and Krom such that for every positive integer n the class contains a sentence φ of a length polynomial in n for which any equivalent BSR sentence contains at least 2^n leading existential quantifiers.*

One possible witness for the mentioned class of MFO sentences consists of all the sentences $\forall x \exists y.\ \bigwedge_{i=1}^{2n} \big(P_i(x) \leftrightarrow Q_i(y)\big)$ for $n \geq 1$.

3.3 Expressiveness of SF

We have already seen that SF is a syntactic extension of MFO and BSR (Proposition 3.1.4) and that every $\mathrm{MFO}_\approx$ sentence is equivalent to some SF sentence (Theorem 3.1.5). Hence, SF is (at least) as expressive as these three fragments. Moreover, any sentence that is a Boolean combination of sentences from BSR and/or $\mathrm{MFO}_\approx$ is equivalent to some SF sentence. On the other hand, Theorem 3.2.7 shows that SF sentences can be considerably more succinct than their BSR equivalents.

3.3.1 Fundamental Properties of Relations

The separated fragment inherits some expressiveness from BSR that other decidable first-order fragments, such as FO^2, FL, and GF, lack. For instance, SF sentences can naturally express fundamental properties of relations, such as reflexivity, irreflexivity, symmetry, antisymmetry, transitivity, and congruence with respect to other predicates. Hence, SF sentences can directly express the axioms of, e.g., equivalence relations, (strict) order relations, and congruence relations. These are fundamental and interesting properties of relations that have to be assumed at the meta-level when dealing with first-order fragments that are less expressive in this respect.

3.3.2 Basic Counting Quantifiers

Basic "there exist at least n" *counting quantifiers* $\exists^{\geq n} y$ can be defined natively in SF and do not have to be introduced via special operators. For example, given some positive integer n, the formula $\exists^{\geq n} y.\, \psi(y, \bar{\mathsf{z}})$ stipulates the existence of n pairwise distinct domain elements $\mathsf{a}_1, \ldots, \mathsf{a}_n$ such that ψ is satisfied if any of the a_i is assigned to y. There is a standard first-order expansion for such formulas, namely

$$\exists y_1 \ldots y_n.\, \bigwedge_{i=1}^{n} \psi(y_i, \bar{\mathsf{z}}) \wedge \bigwedge_{i<j} y_i \not\approx y_j \ .$$

This kind of basic counting quantifiers fits in nicely with the separateness conditions of SF. That is, if all variables that are universally quantified in ψ are separated from all variables that are existentially quantified, including y, then this separateness is preserved in the expansion of $\exists^{\geq n} y.\, \psi(y, \bar{\mathsf{z}})$. The reason is simply that in the expansion the variable y is converted into multiple y_i, each of which is existentially quantified and separated from the universally quantified variables. This would change with the slightly different, yet equivalent, alternative expansion

$$\forall y_1 \ldots y_{n-1} \exists y.\, \psi(y) \wedge \bigwedge_{i=1}^{n-1} y \not\approx y_i \ ,$$

where the existentially quantified y co-occurs with every (universally quantified) y_i in a disequation.

The picture is different, however, for the case of "there exist at most n" counting quantifiers $\exists^{\leq n} x$ with positive n. One possible expansion for any formula $\exists^{\leq n} x.\, \psi(x, \bar{\mathsf{z}})$ is

$$\forall x_1 \ldots x_{n+1}.\, \Big(\bigwedge_{1 \leq i < j \leq n+1} x_i \not\approx x_j \Big) \rightarrow \bigvee_{i=1}^{n+1} \neg \psi(x_i, \bar{\mathsf{z}}) \ .$$

As $\exists^{\leq n} x$ is in a sense dual to $\exists^{\geq n}$, it is not very surprising that its expansion is based on universal quantification. Now consider the formula $\forall u \exists^{\leq 1} y \forall v.\, \neg R(u, v)$ and its equivalent with the expanded counting quantification:

$$\begin{aligned} &\forall u \forall x_1 x_2.\, x_1 \not\approx x_2 \rightarrow \bigvee_{i=1}^{2} \neg \forall v.\, \neg R(u, v) \\ &\models \big(\exists x_1 x_2.\, x_1 \not\approx x_2 \big) \rightarrow \forall u \exists v.\, R(u, v) \ . \end{aligned}$$

Although the original sentence looks rather innocent from the perspective of separateness, its expansion does not. Indeed, the latter can be used as a building block for an *infinity axiom*:

$$\begin{aligned} \varphi_{\text{inf}} := \ & \big(\forall x.\, \neg R(x, x) \big) \\ & \wedge \big(\forall x_1 x_2 x_3.\, R(x_1, x_2) \wedge R(x_2, x_3) \rightarrow R(x_1, x_2) \big) \\ & \wedge \big(\exists y_1 y_2.\, y_1 \not\approx y_2 \big) \\ & \wedge \big(\big(\exists x_1 x_2.\, x_1 \not\approx x_2 \big) \rightarrow \forall u \exists v.\, R(u, v) \big) \ . \end{aligned}$$

Obviously, the hidden negation in any formula $\exists^{\leq n} x.\, \psi(x, \bar{\mathsf{z}})$ makes it a bit tricky to formulate suitable separateness conditions that would allow to integrate such expressions into SF without losing the finite model property (and decidability of SF-Sat). We shall not investigate the counting abilities of SF any further in the present thesis and leave it for future work.

One can use a sentence similar to φ_{inf} and combine it with a formalization of domino problems in SF to obtain a formalization of *unconstrained domino problems*, which are, in general, undecidable — see Section 3.1.1 in [BGG97] for a discussion. In the remainder of this subsection, we shall present such a formalization in SF. It follows that enhancing SF with the described form of "there are at least n" quantifiers renders the associated satisfiability problem undecidable.

Definition 3.3.1 (Unconstrained domino systems, cf. Definition 3.1.2 in [BGG97])**.** *A* domino system $\mathfrak{D} := \langle \mathcal{D}, \mathcal{H}, \mathcal{V} \rangle$ *is a triple where* $\mathcal{D}$ *is a finite set of tiles and* $\mathcal{H}, \mathcal{V} \subseteq \mathcal{D} \times \mathcal{D}$ *are binary relations determining the allowed horizontal and vertical neighbors of tiles, respectively. Consider the space* $\mathbb{N} \times \mathbb{N}$*. We say that* $\mathfrak{D}$ tiles the space $\mathbb{N} \times \mathbb{N}$ *if and only if there exists a mapping* $\tau : \mathbb{N} \times \mathbb{N} \to \mathcal{D}$*, called a* $\mathfrak{D}$-tiling*, such that for every* $\langle x, y \rangle \in \mathbb{N} \times \mathbb{N}$ *the following conditions hold.*

(a) If $\tau(x, y) = D$ *and* $\tau(x + 1, y) = D'$*, then* $\langle D, D' \rangle \in \mathcal{H}$*.*

(b) If $\tau(x, y) = D$ *and* $\tau(x, y + 1) = D'$*, then* $\langle D, D' \rangle \in \mathcal{V}$*.*

Proposition 3.3.2 (Berger [Ber66])**.** *The set of domino systems that tile the space* $\mathbb{N} \times \mathbb{N}$ *forms an undecidable problem.*

In order to formalize a given unconstrained domino problem $\mathfrak{D} = \langle \mathcal{D}, \mathcal{H}, \mathcal{V} \rangle$, we introduce the following constant and predicate symbols:

$\mathrm{Succ}(x, x')$	x' is the successor of x,
$R(x, x')$	R constitutes a strict total order: x is strictly smaller than x',
$H(x, y, x', y')$	$\langle x', y' \rangle$ is the horizontal neighbor of $\langle x, y \rangle$, i.e. x' is successor of x and $y' = y$,
$V(x, y, x', y')$	$\langle x', y' \rangle$ is the vertical neighbor of $\langle x, y \rangle$, i.e. $x' = x$ and y' is successor of y,
$D(x, y)$	$\langle x, y \rangle$ is tiled with $D \in \mathcal{D}$.

First, we stipulate the axioms of the successor relation Succ and of the strict total order R, in which the successor relation is embedded:

$$
\begin{aligned}
\chi_1 &:= \exists z \forall x.\, \neg \mathrm{Succ}(x, z)\ , \\
\chi_2 &:= \forall x \exists x'.\, \mathrm{Succ}(x, x')\ , \\
\chi_3 &:= \forall x x' x''.\, \big(\mathrm{Succ}(x, x') \wedge \mathrm{Succ}(x, x'') \to x' \approx x''\big) \\
&\qquad\qquad \wedge \big(\mathrm{Succ}(x', x) \wedge \mathrm{Succ}(x'', x) \to x' \approx x''\big)\ , \\
\chi_4 &:= \forall x x'.\, \mathrm{Succ}(x, x') \to R(x, x')\ , \\
\chi_5 &:= \forall x.\, \neg R(x, x)\ , \\
\chi_6 &:= \forall x x' x''.\, R(x, x') \wedge R(x', x'') \to R(x, x'')\ , \\
\chi_7 &:= \forall x x'.\, R(x, x') \vee R(x', x)\ .
\end{aligned}
$$

The following sentences encode a given domino system $\mathfrak{D} := \langle \mathcal{D}, \mathcal{H}, \mathcal{V} \rangle$:

$$
\begin{aligned}
\eta_1 &:= \forall x y x' y'.\, H(x, y, x', y') \ \leftrightarrow\ \mathrm{Succ}(x, x') \wedge y \approx y'\ , \\
\eta_2 &:= \forall x y x' y'.\, V(x, y, x', y') \ \leftrightarrow\ x \approx x' \wedge \mathrm{Succ}(y, y')\ , \\
\eta_3 &:= \forall x x' y.\, H(x, y, x', y) \ \to \bigvee_{\langle D, D' \rangle \in \mathcal{H}} \big(D(x, y) \wedge D'(x', y)\big)\ , \\
\eta_4 &:= \forall x y y'.\, V(x, y, x, y') \ \to \bigvee_{\langle D, D' \rangle \in \mathcal{V}} \big(D(x, y) \wedge D'(x, y')\big)\ , \\
\eta_5 &:= \bigwedge_{\substack{D, D' \in \mathcal{D} \\ D \neq D'}} \forall x y.\, D(x, y) \ \to\ \neg D'(x, y)\ .
\end{aligned}
$$

Proposition 3.3.3. *Assume that* $\mathcal{D}$*,* $\mathcal{H}$*, and* $\mathcal{V}$ *are nonempty and let* $\mathcal{A}$ *be a model of the sentence* $\chi_1 \wedge \ldots \wedge \chi_7 \wedge \eta_1 \wedge \ldots \wedge \eta_5$*.* $\mathcal{A}$ *induces a tiling* τ *of* $\mathbb{N} \times \mathbb{N}$*. Conversely, any* $\mathfrak{D}$*-tiling* τ *of the space* $\mathbb{N} \times \mathbb{N}$ *induces a model* $\mathcal{A} \models \chi_1 \wedge \ldots \wedge \chi_7 \wedge \eta_1 \wedge \ldots \wedge \eta_5$*.*

All sentences in the above formalization, except for χ_2, belong to BSR and thus also to SF. We have observed that χ_2 can be expressed with the sentence

$$
\begin{aligned}
&(\exists x_1 x_2.\ x_1 \not\approx x_2) \ \wedge\ \big(\forall u \exists^{\leq 1} y \forall v.\, \neg \mathrm{Succ}(u, v)\big) \\
&\models\!\!\!\models\ \ (\exists x_1 x_2.\ x_1 \not\approx x_2) \ \wedge\ \big((\exists x_1 x_2.\, x_1 \not\approx x_2) \to \forall u \exists v.\, \mathrm{Succ}(u, v)\big)\ .
\end{aligned}
$$

Hence, if we were to admit quantifiers of the form $\exists^{\leq n}$ in SF as described above, then we could reduce undecidable problems to the satisfiability problem of this extended language.

The sentence χ_2 in fact belongs to several of the known decidable fragments: AF, GKS, the Skolem fragment, FO^2, FL, and Maslov's fragment K. Hence, Boolean combinations of BSR sentences with sentences from one of the mentioned other fragments lead to undecidable satisfiability problems, if no further restrictions are imposed. Consequently, it seems impossible that SF could be extended into a decidable fragment that contains BSR and any of the mentioned fragments without seriously restricting the possibility to combine sentences from the extended fragment with Boolean connectives.

3.3.3 Expressiveness with Respect to Models of Bounded Size

Whenever it is possible to restrict our attention to models with a bounded domain size — for some known bound —, then SF is as expressive as full (relational) first-order logic. This alone is not a very interesting result, as already the *existential fragment* of relational first-order logic, represented by the class of relational $\exists^*$ prefix sentences, possesses this property (universal quantification can be replaced by finite conjunctions). What makes the case of SF special is that the incurred blowup in formula length is not linear in the bound but significantly lower.

In order to make this idea more precise, we consider the following formula over the unary predicate symbols $Q_1, \ldots, Q_m$:

χ_m

$$\chi_m := \forall x x'. \Big(\bigwedge_{i=1}^{m} Q_i(x) \leftrightarrow Q_i(x') \Big) \longrightarrow x \approx x' \ .$$

It is easy to see that any model of χ_m contains at most 2^m domain elements. The length of χ_m lies in $\mathcal{O}(m)$.

Proposition 3.3.4. *For every positive integer m and any relational first-order sentence φ in which the predicate symbols $Q_1, \ldots, Q_m$ do not occur there is some SF sentence φ_{SF} such that the sentences $\chi_m \wedge \varphi$ and $\chi_m \wedge \varphi_{\mathrm{SF}}$ are equivalent. Moreover, the length of φ_{SF} lies in $\mathcal{O}\big(m \cdot \mathrm{len}(\varphi)\big)$.*

$s \mathrel{\hat{\approx}}_m t$

For the proof of this result, we use the abbreviation $s \mathrel{\hat{\approx}}_m t := \bigwedge_{i=1}^{m} Q_i(s) \leftrightarrow Q_i(t)$ for any two terms s, t, and we employ the following simple observations.

Lemma 3.3.5. *Let $\mathcal{A}$ be any structure, let β be any variable assignment over $\mathcal{A}$'s domain, and let s, t be two terms. If $\mathcal{A} \models \chi_m$ holds, then we get $\mathcal{A}, \beta \models s \mathrel{\hat{\approx}}_m t$ if and only if $\mathcal{A}, \beta \models s \approx t$.*

This means, if we restrict our attention to domains with at most 2^m domain elements, we can use a separated form of equality.

Proposition 3.3.6. *Let $\psi[t]$ be any formula (first-order or second-order) in which the term t occurs. Let x be some first-order variable that does not occur in $\psi[t]$. Then, $\psi[t]$ is semantically equivalent to $\forall x.\, x \approx t \rightarrow \psi[x]$, where $\psi[x]$ is derived from $\psi[t]$ by replacing every occurrence of t with the variable x.*

Without loss of generality, we assume that φ in Proposition 3.3.4 is in negation normal form. We construct φ_{SF} from φ by consecutively replacing each subformula of the form $\exists y.\, \psi$ in φ with $\exists y \forall v.\, y \mathrel{\hat{\approx}}_m v \rightarrow \psi\big[y/v\big]$, where we assume v to be fresh (one fresh variable for each replaced subformula). This can be done in such a way that any occurrence of existentially quantified variables lies within subformulas $s \mathrel{\hat{\approx}}_m t$. The semantic equivalence of $\chi_m \wedge \varphi$ and $\chi_m \wedge \varphi_{\mathrm{SF}}$ follows from Lemma 3.3.5 and Proposition 3.3.6. This proves Proposition 3.3.4.

Notice that φ_{SF} in fact belongs to the strongly separated fragment (SSF) (cf. Definition 3.2.3). The abbreviation $s \mathrel{\hat{\approx}}_m t$ used above is based on unary predicate symbols. It can only account for exponentially many domain elements in the length of the abbreviated formula, which is linear in the parameter m. This is typical for SSF.

We can replace this approach by a more sophisticated one that makes better use of the potential of SF and allows for more succinct representations. The main conceptual idea is that we use unary

predicates $L_1, \ldots, L_{n-1}$ to define sets $L_1^{\mathcal{A}}, \ldots, L_{n-1}^{\mathcal{A}}$ accompanied with increasing upper bounds on their cardinality $2^m, 2^{2^m}, \ldots, 2^{\uparrow n-1}(m)$. To this end, for all positive m, n and every ℓ with $2 \leq \ell \leq n-1$ we recursively define the abbreviations $u \mathrel{\hat{\approx}}^{\ell}_{m,n} v$

$$u \mathrel{\hat{\approx}}^{1}_{m,1} v := \bigwedge_{i=1}^{m} \big(Q_i(u) \leftrightarrow Q_i(v)\big)$$

and, for $n \geq 2$ we set

$$u \mathrel{\hat{\approx}}^{1}_{m,n} v := L_1(u) \wedge L_1(v) \wedge \bigwedge_{i=1}^{m} \big(Q_i(u) \leftrightarrow Q_i(v)\big)$$

and

$$u \mathrel{\hat{\approx}}^{\ell}_{m,n} v := L_\ell(u) \wedge L_\ell(v) \wedge \Big(\forall x.\, L_{\ell-1}(x) \to \exists y.\, (x \mathrel{\hat{\approx}}^{\ell-1}_{m,n} y) \wedge \big(R_\ell(x,u) \leftrightarrow R_\ell(y,v)\big)\Big)$$

and

$$u \mathrel{\hat{\approx}}^{n}_{m,n} v := \forall x.\, L_{n-1}(x) \to \exists y.\, (x \mathrel{\hat{\approx}}^{n-1}_{m,n} y) \wedge \big(R_n(x,u) \leftrightarrow R_n(y,v)\big)\ .$$

Notice that for every formula $u \mathrel{\hat{\approx}}^{\ell}_{m,n} v$, $1 \leq \ell \leq n$, we can partition $\mathrm{vars}(u \mathrel{\hat{\approx}}^{\ell}_{m,n} v)$ into two disjoint sets X, Y that are separated in the formula, and we have $u \in X$ and $v \in Y$. Moreover, any variable that is universally quantified in $u \mathrel{\hat{\approx}}^{\ell}_{m,n} v$ belongs to X, while Y contains all existentially quantified variables. Also note that no quantifier in the formula occurs within the scope of a negation sign or in the antecedent of an implication. Regarding formula length, we observe $\mathrm{len}\big(u \mathrel{\hat{\approx}}^{1}_{m,n} v\big) \in \mathcal{O}(m)$, and $\mathrm{len}\big(u \mathrel{\hat{\approx}}^{\ell}_{m,n} v\big) \in \mathcal{O}(\ell + m)$ for every ℓ with $2 \leq \ell \leq n$.

Based on these abbreviations for a separated variant of equality, we define the following sentences for all $m \geq 1$ and $n \geq 2$:

$$\chi_{m,1} := \forall x x'.\, (x \mathrel{\hat{\approx}}^{1}_{m,1} x') \to x \approx x' \qquad \chi_{m,k}$$

and

$$\begin{aligned}\chi_{m,n} := {} & \Big(\forall x x'.\, (x \mathrel{\hat{\approx}}^{1}_{m,n} x') \to x \approx x'\Big) \\ & \wedge \Big(\bigwedge_{\ell=2}^{n-1} \forall x x'.\, L_\ell(x) \wedge L_\ell(x') \\ & \qquad \to \exists y \forall y'.\, \Big(\Big(L_{\ell-1}(y) \to (y \mathrel{\hat{\approx}}^{\ell-1}_{m,n} y') \wedge \big(R_\ell(y',x) \leftrightarrow R_\ell(y',x')\big)\Big) \to x \approx x'\Big)\Big) \\ & \wedge \Big(\forall x x' \exists y \forall y'.\, \Big(\Big(L_{n-1}(y) \to (y \mathrel{\hat{\approx}}^{n-1}_{m,n} y') \wedge \big(R_n(y',x) \leftrightarrow R_n(y',x')\big)\Big) \to x \approx x'\Big)\Big)\ .\end{aligned}$$

Due to the syntactic properties of the formulas $u \mathrel{\hat{\approx}}^{\ell}_{m,n} v$ regarding the separateness of variables, the sentences $\chi_{m,n}$ belong to SF, if we transform them into prenex normal form by simply shifting all quantifiers to the front. The only obstacle is the implicit negation sign in front of $(y \mathrel{\hat{\approx}}^{\ell-1}_{m,n} y')$ and similar formulas in the antecedent of implications. However, this neatly fits with the fact that y is existentially quantified and y' universally. Regarding formula length, we observe $\mathrm{len}\big(\chi_{m,1}\big) \in \mathcal{O}(m)$, and $\mathrm{len}\big(\chi_{m,n}\big) \in \mathcal{O}(n^2 + n \cdot m)$ for every $n \geq 2$.

We can extend Lemma 3.3.5 also to the new separated equality $\hat{\approx}^{\ell}_{m,n}$ in the following way.

Lemma 3.3.7. *Let $\mathcal{A}$ be any structure, let β be any variable assignment over $\mathcal{A}$'s domain. Suppose we have $\mathcal{A} \models \chi_{m,n}$ for two integers $m, n \geq 1$. For every ℓ with $1 \leq \ell \leq n-1$ and all first-order variables u, v we get $\mathcal{A}, \beta \models u \mathrel{\hat{\approx}}^{\ell}_{m,n} v$ if and only if $\mathcal{A}, \beta \models u \approx v \wedge L_\ell(u) \wedge L_\ell(v)$. Moreover, we have $\mathcal{A}, \beta \models u \mathrel{\hat{\approx}}^{n}_{m,n} v$ if and only if $\mathcal{A}, \beta \models u \approx v$.*

We next observe that the sentences $\chi^n_{m,n}$ restrict the size of domains.

Lemma 3.3.8. *Let $m, n \geq 1$ and let $\mathcal{A}$ be any model of $\chi_{m,n}$. For every k, $1 \leq k \leq n-1$, the set $L_k^{\mathcal{A}}$ contains at most $2^{\uparrow k}(m)$ domain elements. Moreover, $\mathcal{A}$'s domain contains at most $2^{\uparrow n}(m)$ elements.*

Proof. In case of $\mathcal{A} \models \chi_{m,k}$ with $k = 1$, any two elements $\mathsf{a}, \mathsf{b} \in \mathsf{A}$ that are not distinguishable by their membership in the sets $Q_1^{\mathcal{A}}, \ldots, Q_m^{\mathcal{A}}$ are identical. Hence, A cannot contain more than $2^m = 2^{\uparrow 1}(m)$ distinct elements.

In order to prove the first half of the lemma for $n \geq 2$ under the assumption $\mathcal{A} \models \chi_{m,n}$, we proceed by induction on k, starting with $k = 1$. The base case $k = 1$ is easy to settle, as any two elements $\mathsf{a}, \mathsf{b} \in L_1^{\mathcal{A}}$ that are not distinguishable by their membership in the sets $Q_1^{\mathcal{A}}, \ldots, Q_{m,n}^{\mathcal{A}}$ have to be identical in $\mathcal{A}$. Hence, $L_1^{\mathcal{A}}$ cannot contain more than $2^m = 2^{\uparrow 1}(m)$ distinct elements.

Consider any k with $1 < k \leq n-1$. By induction, the set $L_{k-1}^{\mathcal{A}}$ contains at most $2^{\uparrow k-1}(m)$ elements. The sentence $\chi_{m,n}$ contains the following conjunct for every ℓ with $2 \leq \ell \leq n-1$:

$$\forall x x'.\, L_\ell(x) \wedge L_\ell(x') \rightarrow \exists y \forall y'.\, \Big(\Big(L_{\ell-1}(y) \rightarrow (y \mathrel{\hat{\approx}}_{m,n}^{\ell-1} y') \wedge \big(R_\ell(y', x) \leftrightarrow R_\ell(y', x')\big)\Big) \rightarrow x \approx x'\Big)\,.$$

The intended meaning of this sentence coincides with the following non-separated sentence:

$\psi_{m,\ell}$

$$\begin{aligned}
&\psi_{m,\ell} :=\\
&\forall x x'.\, L_\ell(x) \wedge L_\ell(x') \rightarrow \Big(\Big(\forall y.\, L_{\ell-1}(y) \rightarrow \big(R_\ell(y, x) \leftrightarrow R_\ell(y, x')\big)\Big) \rightarrow x \approx x'\Big)\\
&\quad\models\!\models\ \forall x x'.\, L_\ell(x) \wedge L_\ell(x') \rightarrow \Big(\Big(\forall y.\, L_{\ell-1}(y) \rightarrow \exists y'.\, y \approx y' \wedge \big(R_\ell(y', x) \leftrightarrow R_\ell(y', x')\big)\Big) \rightarrow x \approx x'\Big)\\
&\quad\models\!\models\ \forall x x'.\, L_\ell(x) \wedge L_\ell(x') \rightarrow \exists y \forall y'.\, \Big(\Big(L_{\ell-1}(y) \rightarrow y \approx y' \wedge \big(R_\ell(y', x) \leftrightarrow R_\ell(y', x')\big)\Big) \rightarrow x \approx x'\Big)\,,
\end{aligned}$$

where we could take the last line and replace the equation $y \approx y'$ with its separated variant $y \mathrel{\hat{\approx}}_{m,n}^{\ell-1} y'$ to obtain the above conjunct. By virtue of (a slightly adapted variant of) Lemma 3.3.7, this replacement preserves semantics.

It is easy to see that $\mathcal{A} \models \psi_{m,\ell}$ entails that any two domain elements $\mathsf{a}, \mathsf{b} \in L_\ell^{\mathcal{A}}$ are identical, if the two sets $\big\{\langle \mathsf{a}, \mathsf{c}\rangle \in R_\ell^{\mathcal{A}} \mid \mathsf{c} \in L_{\ell-1}^{\mathcal{A}}\big\}$ and $\big\{\langle \mathsf{b}, \mathsf{c}\rangle \in R_\ell^{\mathcal{A}} \mid \mathsf{c} \in L_{\ell-1}^{\mathcal{A}}\big\}$ coincide. Recall that the inductive hypothesis says that the set $L_{\ell-1}^{\mathcal{A}}$ contains at most $2^{\uparrow \ell-1}(m)$ elements. Hence, $\mathcal{A} \models \psi_{m,\ell}$ entails that the set $L_\ell^{\mathcal{A}}$ contains at most $2^{|L_{\ell-1}^{\mathcal{A}}|} \leq 2^{2^{\uparrow \ell-1}(m)} = 2^{\uparrow \ell}(m)$ domain elements.

Finally, the conjunct

$$\forall x x' \exists y \forall y'.\, \Big(\Big(L_{n-1}(y) \rightarrow (y \mathrel{\hat{\approx}}_{m,n}^{n-1} y') \wedge \big(R_n(y', x) \leftrightarrow R_n(y', x')\big)\Big) \rightarrow x \approx x'\Big)$$

in the sentence $\chi_{m,n}$ has the following intended meaning:

$$\forall x x'.\, \Big(\Big(\forall y.\, L_{n-1}(y) \rightarrow \big(R_n(y, x) \leftrightarrow R_n(y, x')\big)\Big) \rightarrow x \approx x'\Big)\,.$$

As we have already shown that $\mathcal{A} \models \chi_{m,n}$ entails $|L_{n-1}^{\mathcal{A}}| \leq 2^{\uparrow n-1}(m)$, we conclude that $\mathcal{A} \models \chi_{m,n}$ also implies that $\mathcal{A}$'s domain A contains at most $2^{\uparrow n}(m)$ elements. □

As a counterpart to the upper bound result in Lemma 3.3.8, we observe that $\chi_{m,n}$ does not restrict the cardinality of models further than this.

Lemma 3.3.9. *Let $\mathcal{A}$ be any structure whose domain contains at most $2^{\uparrow n}(m)$ elements. There is a model $\mathcal{A}' \models \chi_{m,n}$ over the same domain that differs from $\mathcal{A}$ only in its interpretation of the predicate symbols $L_1, \ldots, L_{n-1}, R_2, \ldots, R_n, Q_1, \ldots, Q_m$.*

We use Lemmas 3.3.8 ans 3.3.9 to derive a much stronger variant of Proposition 3.3.4. Abstractly speaking, it states that, when restricted to models of the size $2^{\uparrow n}(m)$, any first-order sentence can be translated into an equisatisfiable SF sentence whose length is polynomial in n, m, and the length of the original sentence.

Lemma 3.3.10. *Let m, n be two positive integers with $m \geq 1$ and $n \geq 2$. There exists an effective translation $T_{m,n}$ mapping relational first-order sentences φ to SF sentences φ_{SF} that satisfy the following properties. For every relational sentence φ, which does not contain the predicate symbols*

$T_{m,n}$

$L_1, \dots, L_{n-1}, R_2, \dots, R_n, Q_1, \dots, Q_m$, we have
(a) $\chi_{m,n} \wedge \varphi \models\!\!\!\models \chi_{m,n} \wedge T_{m,n}(\varphi)$,
(b) the formula length of $T_{m,n}(\varphi)$ *is at most* $p(m,n) \cdot \mathrm{len}(\varphi)$ *for some polynomial* $p(m,n)$*, and*
(c) $T_{m,n}(\varphi)$ *is computable in time* $q(m, n, \mathrm{len}(\varphi))$ *for some polynomial* $q(m,n,k)$.

Proof. The translation $T_{m,n}$ is very similar to the one we have already sketched above. First, we transform φ into negation normal form. We construct φ_{SF} from φ by consecutively replacing each subformula of the form $\exists y.\, \psi$ in φ with $\exists y \forall v.\, y \mathrel{\hat{\approx}}^n_{m,n} v \to \psi\big[y/v\big]$, where we assume v to be fresh (one fresh variable for each replaced subformula). Finally, all quantifiers are shifted to the front of the sentence.

We have observed earlier that any variable set $\mathrm{vars}(y \mathrel{\hat{\approx}}^n_{m,n} v)$ can be partitioned into two sets X, Y that are separated in the subformula $y \mathrel{\hat{\approx}}^n_{m,n} v$, where all universally quantified variables are collected in X and all existentially quantified variables belong to Y. Since the introduced subformulas $y \mathrel{\hat{\approx}}^n_{m,n} v$ always occur in the antecedent of implications and are, hence, subject to one implicit negation, the constructed sentence φ_{SF} is indeed an SF sentence. The semantic equivalence of the formulas $\chi_{m,n} \wedge \varphi$ and $\chi_{m,n} \wedge \varphi_{\mathrm{SF}}$ follows from Proposition 3.3.6 and Lemma 3.3.7. □

Lemma 3.3.10 has interesting consequences. For instance, concerning the computational hardness of SF's satisfiability problem. The following theorem entails that SF-Sat is computationally at least as hard as the satisfiability problem for any first-order fragment that enjoys a small model property with an elementary upper bound on the size of small models. For instance, the fragments AF, GKS, FO^2, and GF fall into this category. Even the satisfiability problem for first-order fragments enjoying a small model property with bounds $2^{\uparrow\lceil c \cdot \mathrm{len}(\varphi)\rceil}(\lceil d \cdot \mathrm{len}(\varphi)\rceil)$ for constants c, d, such as FL, can be polynomially reduced to SF-Sat. Although this latter observation already yields a non-elementary lower bound regarding the computational complexity of SF-Sat, we shall derive a more accurate lower bound in Section 5.3 by encoding *bounded domino problems*.

Theorem 3.3.11. *Consider any nonempty class $\mathcal{C}$ of relational first-order sentences for which we know two constants $c, d \geq 1$ such that every satisfiable φ in $\mathcal{C}$ has a model whose domain contains at most $2^{\uparrow\lceil c \cdot \mathrm{len}(\varphi)\rceil}(\lceil d \cdot \mathrm{len}(\varphi)\rceil)$ elements. The satisfiability problem for $\mathcal{C}$ is polynomial-time reducible to SF-Sat.*

Proof. We use the translations $T_{m,n}$ from Lemma 3.3.10 for the reduction from $\mathcal{C}$'s satisfiability problem to SF-Sat. Given any sentence φ from $\mathcal{C}$, we compute $m := \lceil d \cdot \mathrm{len}(\varphi)\rceil$ and $n := \lceil c \cdot \mathrm{len}(\varphi)\rceil$. Without loss of generality, we assume that φ does not contain any of the predicate symbols $L_1, \dots, L_{n-1}, R_2, \dots, R_n, Q_1, \dots, Q_m$. Next, we construct the sentence $\chi_{m,n} \wedge T_{m,n}(\varphi)$. By Lemma 3.3.10, this can be done in time that is polynomial in $\mathrm{len}(\varphi)$ and thus also polynomial in $\|\varphi\|$. By Lemma 3.3.9, any model $\mathcal{A} \models \varphi$ whose domain contains at most $2^{\uparrow\lceil c \cdot \mathrm{len}(\varphi)\rceil}(\lceil d \cdot \mathrm{len}(\varphi)\rceil)$ elements can be extended to a model $\mathcal{A}' \models \chi_{m,n} \wedge \varphi$ over the same domain. Hence, if φ is satisfiable, then there is some model $\mathcal{A}' \models \chi_{m,n} \wedge \varphi$ with $|\mathsf{A}'| \leq 2^{\uparrow\lceil c \cdot \mathrm{len}(\varphi)\rceil}(\lceil d \cdot \mathrm{len}(\varphi)\rceil)$. By Lemma 3.3.10, $\mathcal{A}'$ is also a model of $\chi_{m,n} \wedge T_{m,n}(\varphi)$. On the other hand, any model of $\chi_{m,n} \wedge T_{m,n}(\varphi)$ yields a model of φ. □

Remark 3.3.12. *The restriction of Theorem 3.3.11 to classes over relational vocabularies is not essential. It is folklore knowledge that every first-order sentence containing function symbols can be converted into an equisatisfiable sentence over some relational vocabulary. The function symbols are replaced with predicate symbols that represent the respective function graph. This conversion causes a blowup that is only linear in the length of the original formula.*

For example, the sentence $\forall x.\, P(f(x)) \vee f(x) \approx c$ with function symbols f and c is converted into the equisatisfiable sentence

$$\begin{aligned}
&\big(\forall xyz.\, Q_c(z) \wedge Q_f(x,y) \to P(y) \vee y \approx z\big) \\
&\wedge \big(\exists z.\, Q_c(z)\big) \wedge \big(\forall uv.\, Q_c(u) \wedge Q_c(v) \to u \approx v\big) \\
&\wedge \big(\forall u \exists w.\, Q_f(u,w)\big) \wedge \big(\forall uvw.\, Q_f(u,v) \wedge Q_c(u,w) \to v \approx w\big)\ .
\end{aligned}$$

Employing the ideas underlying Lemma 3.3.10, one can also derive other lower bounds regarding the length of sentences that are equivalent to SF sentences but adhere to certain syntactic restrictions. We have already seen a result in this direction in Section 3.2, namely, Theorem 3.2.7, which described a non-elementary gap between the length of SF sentences and shortest equivalent BSR sentences. A classical result by Gaifman [Gai82] states that every first-order formula is equivalent to some formula that is *local* in a certain sense (see below). It has been shown later [DGKS07a] that there is a non-elementary gap between the length of first-order sentences and their shortest equivalents in *Gaifman normal form*. We intend to prove that this gap also applies to the separated fragment. But first we need some preliminary definitions, mainly taken over from [DGKS07a] (see also the textbooks [EF99, Lib04]).

$\mathcal{G}_\mathcal{A}$

Fix any relational vocabulary Σ and let $\mathcal{A}$ be any Σ-structure. The *Gaifman graph of* $\mathcal{A}$ is the undirected, loop-free graph $\mathcal{G}_\mathcal{A}$ over the vertex set A and the edge set E that satisfies the following property. The set E contains an edge $\langle \mathsf{a}, \mathsf{b}\rangle$ if and only if there is some m-ary predicate symbol P in Σ, some tuple $\langle \mathsf{c}_1, \ldots, \mathsf{c}_m\rangle \in P^\mathcal{A}$, and two distinct indices i, j with $\mathsf{a} = \mathsf{c}_i$ and $\mathsf{b} = \mathsf{c}_j$. The

$\mathrm{dist}_\mathcal{A}(\mathsf{a}, \mathsf{b})$

distance between two domain elements $\mathsf{a}, \mathsf{b} \in \mathsf{A}$ in $\mathcal{A}$ is denoted by $\mathrm{dist}_\mathcal{A}(\mathsf{a}, \mathsf{b})$ and is defined to be the length of the shortest path from a to b in $\mathcal{G}_\mathcal{A}$ — the length of a path is the number of edges on the path. For every nonnegative integer r and every domain element $\mathsf{a} \in \mathsf{A}$ the *r-neighborhood of* a *in* $\mathcal{A}$ is the set $\{\mathsf{b} \in \mathsf{A} \mid \mathrm{dist}_\mathcal{A}(\mathsf{a}, \mathsf{b}) \leq r\}$. The substructure of $\mathcal{A}$ induced by this set is denoted by

$\mathcal{N}^r_\mathcal{A}(\mathsf{a})$

$\mathcal{N}^r_\mathcal{A}(\mathsf{a})$.

For every nonnegative integer r let $\mathrm{dist}_{>r}(x, y)$ be a first-order formula stipulating that the distance between x and y is at least $r+1$. A first-order formula $\psi(x)$ is *r-local* if for every structure

basic local sentence

$\mathcal{A}$ and every $\mathsf{a} \in \mathsf{A}$ we have $\mathcal{A} \models \psi(\mathsf{a})$ if and only if $\mathcal{N}^r_\mathcal{A}(\mathsf{a}) \models \psi(\mathsf{a})$. A *basic local sentence* is a sentence of the form

$$\exists x_1 \ldots x_k. \bigwedge_{1 \leq i < j \leq k} \mathrm{dist}_{>2r}(x_i, x_j) \wedge \bigwedge_{1 \leq i \leq k} \psi(x_i) \ ,$$

Gaifman normal form

where $\psi(x)$ is r-local. A first-order sentence φ is said to be in *Gaifman normal form* if it is a Boolean combination of basic local sentences.

Proposition 3.3.13 (Gaifman [Gai82])**.** *Every relational first-order sentence is equivalent to some first-order sentence in Gaifman normal form.*

Dawar et al. [DGKS07a] present a class of first-order sentences φ that are non-elementarily more succinct than the shortest equivalent sentences φ' in Gaifman normal form. The main tool for the proof is an encoding of nonnegative integers by trees, introduced in Section 10.3 of [FG06], and succinct first-order formulas for handling these trees.

Definition 3.3.14 (Encoding integers by trees (adapted from [DGKS07a], Definition 1))**.** *For*

$\mathrm{bit}(i, n)$

nonnegative integers i, n we write $\mathrm{bit}(i, n)$ *to denote the i-th bit in the binary representation of*

$\mathcal{T}(i)$

n; the least significant bit is $\mathrm{bit}(0, n)$*. We define the tree representation $\mathcal{T}(n)$ for any integer n inductively as follows. $\mathcal{T}(0)$ is the tree consisting only of the root node. For $n > 0$ the tree $\mathcal{T}(n)$ is obtained by creating a new root node and attaching to it all trees $\mathcal{T}(i)$ for which* $\mathrm{bit}(i, n) = 1$.

In Figure 3.1 the tree encoding is illustrated. Notice that the number of sons of the root node equals the number 1-bits in the binary representation of the encoded integer. Further examples can be found in [FG06], page 251.

The height of a tree $\mathcal{T}(n)$ and the number n represented by it are related as follows.

Proposition 3.3.15 ([FG06], Lemma 10.20)**.** *For every tree T let* $\mathrm{height}(T)$ *be the number of edges along the longest path from the root of T to any leaf in T. Then, for all nonnegative integers h, n we have* $\mathrm{height}(\mathcal{T}(n)) \leq h$ *if and only if* $n < 2^{\uparrow h}(1)$.

In addition, we establish the following upper bound regarding the total number of nodes in any tree $\mathcal{T}(n)$.

$h(n)$

Lemma 3.3.16. *For every nonnegative n let $h(n) := \mathrm{height}(\mathcal{T}(n))$. The number of nodes in a tree $\mathcal{T}(n)$ is at most $2^{\uparrow h(n)+1}(1)$.*

Figure 3.1: Left-hand side: the tree $\mathcal{T}(10)$. Right-hand side: the tree $\mathcal{T}(67)$.

Proof. Whenever we refer to trees in this proof, we mean trees that encode integers in the sense of Definition 3.3.14. Fix any nonnegative integer n. Let $h := h(n)$. We proceed by induction on h.

Base cases $h = 0$ and $h = 1$. The only tree with height 0 is $\mathcal{T}(0)$, which contains $1 \leq 2 = 2^{\uparrow 1}(1)$ nodes. Moreover, there is only one tree of height 1, namely $\mathcal{T}(1)$. This tree contains $2 \leq 4 = 2^{\uparrow 2}(1)$ nodes.

Inductive case $h > 1$. For every $h' \geq 0$ let $\text{trees}(h')$ denote the number of distinct trees of height h' and let $\text{nodes}(h')$ denote the maximal number of nodes in any tree of height h'. We observe the following:

$$\begin{aligned}
\text{nodes}(h) &\leq 1 + \sum_{i=0}^{h-1} \text{trees}(i) \cdot \text{nodes}(i) \\
&\leq 1 + \underbrace{\text{nodes}(h-1)}_{\substack{\text{IH} \\ \leq 2^{\uparrow h}(1)}} \cdot \underbrace{\sum_{i=0}^{h-1} \text{trees}(i)}_{= 2^{\uparrow h-1}(1)} \\
&\leq 1 + 2^{\uparrow h}(1) \cdot 2^{\uparrow h-1}(1) \\
&\leq 2^{2 \cdot 2^{\uparrow h-1}(1)} \\
&\leq 2^{2^{\uparrow h}(1)} = 2^{\uparrow h+1}(1)
\end{aligned}$$

□

One of the main results in [DGKS07a] is the following lower bound regarding the length of sentences in Gaifman normal form.

Proposition 3.3.17 ([DGKS07a], Theorem 2)**.** *Let $\Sigma := \langle \{E\}, \emptyset \rangle$ be a vocabulary where E is a binary predicate symbol. For every $h \geq 1$ there is a first-order Σ-sentence φ_h of length $\mathcal{O}(h^4)$ such that every first-order Σ-sentence in Gaifman normal form that is equivalent to φ_h on the class $\mathcal{F}_{\leq h}$ of finite forests (of pairwise distinct trees) of height at most h has length at least $2^{\uparrow h}(1)$.* $\mathcal{F}_{\leq h}$

Although there does not seem to be an obvious way to transfer this result to the realm of SF by straightforward application of Lemma 3.3.10, the underlying ideas facilitate the derivation of a similar lower bound for SF.

Theorem 3.3.18. *There is some vocabulary Σ and some polynomial $p(h)$ such that for every $h \geq 0$ there is an SF Σ-sentence $\varphi_{\text{SF},h}$ of length $p(h)$ satisfying the following property. Every first-order Σ-sentence ψ in Gaifman normal form that is equivalent to $\varphi_{\text{SF},h}$ has length at least $2^{\uparrow h}(1)$.*

Proof sketch. Let $\Sigma := \langle \Pi, \emptyset \rangle$ be the vocabulary where Π contains the unary predicate symbols $\text{Red}, \text{Blue}, Q_1^{\text{r}}, Q_1^{\text{b}}, Q_2^{\text{r}}, Q_2^{\text{b}}, \ldots$ and $L_1^{\text{r}}, L_1^{\text{b}}, L_2^{\text{r}}, L_2^{\text{b}}, \ldots$ and the binary predicate symbols E and $R_1^{\text{r}}, R_1^{\text{b}}, R_2^{\text{r}}, R_2^{\text{b}}, \ldots$ and no further symbols. The superscript r stands for *red* and b stands for *blue*.

In what follows, we abbreviate the expression $2^{\uparrow n}(1)$ with $2^{\uparrow n}$ for any positive integer n. Given $2^{\uparrow n}$ some Σ-structure $\mathcal{A}$, a domain element $\mathsf{a} \in \mathcal{A}$ is considered to be *red* if $\mathsf{a} \in \text{Red}^{\mathcal{A}}$; it is considered

blue if $\mathsf{a} \in \mathrm{Blue}^{\mathcal{A}}$; and it is considered *black* if it is neither red nor blue. We call any given subset of $\mathcal{A}$'s domain red, blue, or black, if all of its elements have the respective color. The predicate symbol E serves as the edge relation for directed graphs, in particular forests of rooted trees.

$\mathcal{CF}_h$ For every positive integer h let $\mathcal{CF}_h$ be the set of Σ-structures $\mathcal{F}_{h,k}$ which we define as follows.
$\mathcal{F}_{h,k}$ For all nonnegative integers h, k with $h \geq 1$ and $k \geq 0$, we define the structure $\mathcal{F}_{h,k}$ to contain the trees $\mathcal{T}(0), \ldots, \mathcal{T}(2^{\uparrow h+2} - 1)$, each colored entirely red or blue; none of the nodes in any tree in $\mathcal{F}_{h,k}$ is red and blue at the same time. Every tree in $\mathcal{F}_{h,k}$, except for $\mathcal{T}(k)$, is colored blue; $\mathcal{T}(k)$ is red. Every tree occurs at most once, no matter its color, e.g. if there is a red tree $\mathcal{T}(\ell)$, then there is neither another red tree $\mathcal{T}(\ell)$ nor a blue $\mathcal{T}(\ell)$ in $\mathcal{F}_{h,k}$. In addition to the trees, there are two disjoint sets, each consisting of $2^{\uparrow h+3}$ pairwise distinct black domain elements $\mathsf{a}_1, \ldots, \mathsf{a}_{2^{\uparrow h+3}} \in \mathsf{F}_{h,k}$ and $\mathsf{b}_1, \ldots, \mathsf{b}_{2^{\uparrow h+3}} \in \mathsf{F}_{h,k}$, none of which occurs in any edge in $\mathcal{F}_{h,k}$'s edge relation $E^{\mathcal{F}_{h,k}}$. We do *not* consider these $\mathsf{a}_i, \mathsf{b}_j$ as trees $\mathcal{T}(0)$. Figure 3.2 illustrates the exemplary structure $\mathcal{F}_{0,1}$.

Across all the structures in the class $\mathcal{CF}_h$ we fix some interpretation of the predicate symbols L_i^{b}, L_i^{r}, Q_i^{b}, Q_i^{r}, R_i^{b}, and R_i^{r} and make sure that the following restrictions are obeyed. These predicates will be used in the same manner as we have already used them previously — the $L_i^{\mathrm{b}}, Q_i^{\mathrm{b}}, R_i^{\mathrm{b}}$ will be used to enforce an upper bound regarding the size of the blue part of $\mathcal{F}_{h,k}$ and for identifying blue elements in a separated fashion, and the $L_i^{\mathrm{r}}, Q_i^{\mathrm{r}}, R_i^{\mathrm{r}}$ will serve the same purpose in the red part of $\mathcal{F}_{h,k}$. For every $\mathcal{F}_{h,k} \in \mathcal{CF}_h$ we assume the following properties.

- For every ℓ, $1 \leq \ell \leq h+3$, we have $\mathsf{a}_1, \ldots, \mathsf{a}_{2^{\uparrow \ell}} \in L_\ell^{\mathrm{b}\mathcal{F}_{h,k}}$ and $\mathsf{b}_1, \ldots, \mathsf{b}_{2^{\uparrow \ell}} \in L_\ell^{\mathrm{r}\mathcal{F}_{h,k}}$.
- For all distinct $\mathsf{a}_i, \mathsf{a}_{i'} \in L_1^{\mathrm{b}\mathcal{F}_{h,k}}$ there is some Q_j^{b} with $\mathsf{a}_i \in Q_j^{\mathrm{b}\mathcal{F}_{h,k}}$ and $\mathsf{a}_{i'} \notin Q_j^{\mathrm{b}\mathcal{F}_{h,k}}$ or vice versa. The analogous property shall hold for distinct $\mathsf{b}_i, \mathsf{b}_{i'} \in L_1^{\mathrm{r}\mathcal{F}_{h,k}}$ and the sets $Q_j^{\mathrm{r}\mathcal{F}_{h,k}}$.
- For every ℓ, $2 \leq \ell \leq h+3$, and all distinct $\mathsf{a}_i, \mathsf{a}_{i'} \in L_\ell^{\mathrm{b}\mathcal{F}_{h,k}}$ there is some element $\mathsf{a}' \in L_{\ell-1}^{\mathrm{b}\,\mathcal{F}_{h,k}}$ with $\langle \mathsf{a}', \mathsf{a}_i \rangle \in R_\ell^{\mathrm{b}\mathcal{F}_{h,k}}$ and $\langle \mathsf{a}', \mathsf{a}_{i'} \rangle \notin R_\ell^{\mathrm{b}\mathcal{F}_{h,k}}$ or vice versa. The analogous property shall hold for distinct $\mathsf{b}_i, \mathsf{b}_{i'} \in L_\ell^{\mathrm{r}\mathcal{F}_{h,k}}$, some element $\mathsf{b}' \in L_{\ell-1}^{\mathrm{r}\,\mathcal{F}_{h,k}}$, and the set $R_\ell^{\mathrm{r}\mathcal{F}_{h,k}}$.
- For any two distinct tree nodes $\mathsf{c}, \mathsf{d} \in \mathrm{Blue}^{\mathcal{F}_{h,k}}$ there is some $\mathsf{a}_i \in L_{h+2}^{\mathrm{b}\,\mathcal{F}_{h,k}}$ with $\langle \mathsf{a}_i, \mathsf{c} \rangle \in R_\ell^{\mathrm{b}\mathcal{F}_{h,k}}$ and $\langle \mathsf{a}_i, \mathsf{d} \rangle \notin R_\ell^{\mathrm{b}\mathcal{F}_{h,k}}$ or vice versa. The analogous property shall hold for all distinct red tree nodes c, d, some element $\mathsf{b}_i \in L_{h+2}^{\mathrm{r}\,\mathcal{F}_{h,k}}$, and the set $R_\ell^{\mathrm{r}\mathcal{F}_{h,k}}$.
- In the Gaifman graph $\mathcal{G}_{\mathcal{F}_{h,k}}$ the blue part of $\mathcal{F}_{h,k}$ is disconnected from the red part of $\mathcal{F}_{h,k}$. There are no connections between (1) any a_i and any b_j, (2) any a_i and any $\mathsf{d} \in \mathrm{Red}^{\mathcal{F}_{h,k}}$, (3) any b_j and any $\mathsf{c} \in \mathrm{Blue}^{\mathcal{F}_{h,k}}$, (4) any $\mathsf{c} \in \mathrm{Blue}^{\mathcal{F}_{h,k}}$ and any $\mathsf{d} \in \mathrm{Red}^{\mathcal{F}_{h,k}}$.

Moreover, we assume that for all distinct $\mathcal{F}_{h,k}, \mathcal{F}_{h,k'} \in \mathcal{CF}_h$ the following properties hold.

(a) $\mathcal{F}_{h,k}$ and $\mathcal{F}_{h,k'}$ have the same domain, i.e. $\mathsf{F}_{h,k} = \mathsf{F}_{h,k'}$.

(b) The two substructures $\mathcal{F}'_{h,k}, \mathcal{F}'_{h,k'}$ of $\mathcal{F}_{h,k}$ and $\mathcal{F}_{h,k'}$ induced by the set $\{\mathsf{a}_1, \ldots, \mathsf{a}_{2^{\uparrow h+2}}, \mathsf{b}_1, \ldots, \mathsf{b}_{2^{\uparrow h+2}}\}$, respectively, coincide.

(c) Consider any node t in any tree $\mathcal{T}(\ell)$ and let $\mathsf{c} \in \mathsf{F}_{h,k}$ and $\mathsf{d} \in \mathsf{F}_{h,k'}$ be the domain elements corresponding to this node t in the respective structure.

- If $\ell = k$, then for every i, $1 \leq i \leq 2^{\uparrow h+3}$, we have $\mathcal{F}_{h,k} \models R_{h+4}^{\mathrm{r}}(\mathsf{b}_i, \mathsf{c})$ if and only if $\mathcal{F}_{h,k'} \models R_{h+4}^{\mathrm{b}}(\mathsf{a}_i, \mathsf{d})$.
- If $\ell = k'$, then for every i, $1 \leq i \leq 2^{\uparrow h+3}$, we have $\mathcal{F}_{h,k} \models R_{h+4}^{\mathrm{b}}(\mathsf{a}_i, \mathsf{c})$ if and only if $\mathcal{F}_{h,k'} \models R_{h+4}^{\mathrm{r}}(\mathsf{b}_i, \mathsf{d})$.
- If $\ell \neq k, k'$, then for every i, $1 \leq i \leq 2^{\uparrow h+3}$, we have $\mathcal{F}_{h,k} \models R_{h+4}^{\mathrm{b}}(\mathsf{a}_i, \mathsf{c})$ if and only if $\mathcal{F}_{h,k'} \models R_{h+4}^{\mathrm{b}}(\mathsf{a}_i, \mathsf{d})$.

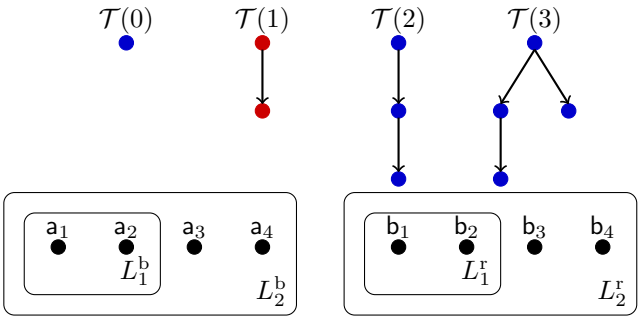


Figure 3.2: Illustration of the structure $\mathcal{F}_{0,1}$.

We now start constructing the sentences $\varphi_{\mathrm{SF},h}$ for any $h \geq 0$. First, we create two variants of the abbreviations $u \mathrel{\hat{\approx}}^{\ell}_{m,n} v$, namely a red variant $u \mathrel{\hat{\approx}}^{\mathrm{r},\ell}_{m,n} v$ and a blue variant $u \mathrel{\hat{\approx}}^{\mathrm{b},\ell}_{m,n} v$, where all Q_i, L_i, R_i are replaced by $Q_i^{\mathrm{r}}, L_i^{\mathrm{r}}, R_i^{\mathrm{r}}$ and $Q_i^{\mathrm{b}}, L_i^{\mathrm{b}}, R_i^{\mathrm{b}}$, respectively, and the recursive reference to $x \mathrel{\hat{\approx}}^{\ell'}_{m,n}$ is replaced by $x \mathrel{\hat{\approx}}^{\mathrm{r},\ell'}_{m,n} y$ and $x \mathrel{\hat{\approx}}^{\mathrm{b},\ell'}_{m,n} y$, respectively. This replacement does not significantly change the length of the formulas, i.e. $\mathrm{len}\big(u \mathrel{\hat{\approx}}^{\mathrm{b},\ell}_{m,n} v\big)$ and $\mathrm{len}\big(u \mathrel{\hat{\approx}}^{\mathrm{r},\ell}_{m,n} v\big)$ are polynomial in m and ℓ. Based on these new formulas, we define variants of the sentences $\chi_{m,n}$ with $n \geq 2$ as follows:

$u \mathrel{\hat{\approx}}^{\mathrm{r},\ell}_{m,n} v$, $u \mathrel{\hat{\approx}}^{\mathrm{b},\ell}_{m,n} v$

$\chi^{\mathrm{r}}_{m,n}$, $\chi^{\mathrm{b}}_{m,n}$

$$\begin{aligned}
\chi^{\mathrm{r}}_{m,n} := {} & \Big(\forall x x'.\, (x \mathrel{\hat{\approx}}^{\mathrm{r},1}_{m,n} x') \to x \approx x'\Big) \\
& \wedge \Big(\bigwedge_{\ell=2}^{n-1} \forall x x'.\, L^{\mathrm{r}}_{\ell}(x) \wedge L^{\mathrm{r}}_{\ell}(x') \\
& \qquad \to \exists y \forall y'.\, \Big(\Big(L^{\mathrm{r}}_{\ell-1}(y) \to (y \mathrel{\hat{\approx}}^{\mathrm{r},\ell-1}_{m,n} y') \wedge \big(R^{\mathrm{r}}_{\ell}(y', x) \leftrightarrow R^{\mathrm{r}}_{\ell}(y', x')\big)\Big) \to x \approx x'\Big)\Big) \\
& \wedge \Big(\forall x x'.\, \mathrm{Red}(x) \wedge \mathrm{Red}(x') \\
& \qquad \to \exists y \forall y'.\, \Big(\Big(L^{\mathrm{r}}_{n-1}(y) \to (y \mathrel{\hat{\approx}}^{\mathrm{r},n-1}_{m,n} y') \wedge \big(R^{\mathrm{r}}_{n}(y', x) \leftrightarrow R^{\mathrm{r}}_{n}(y', x')\big)\Big) \to x \approx x'\Big)\Big)
\end{aligned}$$

and

$$\begin{aligned}
\chi^{\mathrm{b}}_{m,n} := {} & \Big(\forall x x'.\, (x \mathrel{\hat{\approx}}^{\mathrm{b},1}_{m,n} x') \to x \approx x'\Big) \\
& \wedge \Big(\bigwedge_{\ell=2}^{n-1} \forall x x'.\, L^{\mathrm{b}}_{\ell}(x) \wedge L^{\mathrm{b}}_{\ell}(x') \\
& \qquad \to \exists y \forall y'.\, \Big(\Big(L^{\mathrm{b}}_{\ell-1}(y) \to (y \mathrel{\hat{\approx}}^{\mathrm{b},\ell-1}_{m,n} y') \wedge \big(R^{\mathrm{b}}_{\ell}(y', x) \leftrightarrow R^{\mathrm{b}}_{\ell}(y', x')\big)\Big) \to x \approx x'\Big)\Big) \\
& \wedge \Big(\forall x x'.\, \mathrm{Blue}(x) \wedge \mathrm{Blue}(x') \\
& \qquad \to \exists y \forall y'.\, \Big(\Big(L^{\mathrm{b}}_{n-1}(y) \to (y \mathrel{\hat{\approx}}^{\mathrm{b},n-1}_{m,n} y') \wedge \big(R^{\mathrm{b}}_{n}(y', x) \leftrightarrow R^{\mathrm{b}}_{n}(y', x')\big)\Big) \to x \approx x'\Big)\Big).
\end{aligned}$$

Compared to the original $\chi_{m,n}$, the changes are basically the same as the changes made in the abbreviations $u \mathrel{\hat{\approx}}^{\mathrm{r},\ell}_{m,n} v$ and $u \mathrel{\hat{\approx}}^{\mathrm{b},\ell}_{m,n} v$. Moreover, in the third conjunct of $\chi^{\mathrm{r}}_{m,n}$ and $\chi^{\mathrm{b}}_{m,n}$ the antecedents $\mathrm{Red}(x) \wedge \mathrm{Red}(x')$ and $\mathrm{Blue}(x) \wedge \mathrm{Blue}(x')$ are added. The result is that $\chi^{\mathrm{r}}_{m,n}$ does not restrict the whole domain of any model $\mathcal{A} \models \chi^{\mathrm{r}}_{m,n}$ to a certain cardinality, but only the number of domain elements that belong to the set $\mathrm{Red}^{\mathcal{A}}$. A similar effect applies to the sentence $\chi^{\mathrm{b}}_{m,n}$ with respect to the set $\mathrm{Blue}^{\mathcal{A}}$. Regarding formula length, we observe that both $\mathrm{len}\big(\chi^{\mathrm{b}}_{m,n}\big)$ and $\mathrm{len}\big(\chi^{\mathrm{r}}_{m,n}\big)$ lie in $\mathcal{O}(n^2 + n \cdot m)$ for every $n \geq 2$.

We borrow the following formulas from [DGKS07b], the full version of [DGKS07a], listed with their intended meaning:

$\mathrm{eq}_h(x, y)$	the subtrees with roots x and y, respectively, represent the same tree $\mathcal{T}(m)$ for some $m < 2^{\uparrow h}$,

$\text{encoding}_h(x)$	the element x is indeed the root of a subtree $\mathcal{T}(m)$ for some $m < 2^{\uparrow h}$,
$\text{less}_h(x,y)$	the subtree with root x represents a tree $\mathcal{T}(m)$ for some $m < 2^{\uparrow h}$ and the subtree with root y represents a tree $\mathcal{T}(m')$ for some m' with $m < m' < 2^{\uparrow h}$,
$\min(x)$	the subtree with root x represents the tree $\mathcal{T}(0)$,
$\max_h(x)$	the subtree with root x represents the tree $\mathcal{T}\big(2^{\uparrow h} - 1\big)$,
$\text{succ}_h(x,y)$	for the two subtrees T_x, T_y with root x, y, respectively, we have that if T_x represents the tree $\mathcal{T}(m)$ with $m < 2^{\uparrow h} - 1$, then T_y represents the tree $T(m+1)$.

The following formula definitions are taken over from [FG06], Section 10.3 ($\text{eq}_0(x,y)$ and $\text{eq}_h(x,y)$), and from [DGKS07b] ($\text{encoding}_h(x)$, $\text{less}_h(x,y)$, $\min(x)$, $\max_h(x)$, and $\text{succ}_h(x,y)$). The definitions containing the parameter h are meant for positive h.

$$
\begin{aligned}
\text{eq}_0(x,y) &:= \texttt{true} \\
\text{eq}_h(x,y) &:= \big(\big(\exists u.\, E(x,u)\big) \leftrightarrow \big(\exists v.\, E(y,v)\big)\big) \\
&\quad \wedge \Big(\forall w.\, E(x,w) \rightarrow \Big(\exists z.\, E(y,z) \wedge \Big(\forall z'.\, E(y,z') \rightarrow \Big(\exists w'.\, E(x,w') \wedge \\
&\qquad \Big(\forall uv.\big(u \approx w \wedge v \approx z\big) \vee \big(u \approx w' \wedge v \approx z'\big) \rightarrow \text{eq}_{h-1}(u,v)\Big)\Big)\Big)\Big)\Big) \\
\text{encoding}_0(x) &:= \forall x'.\, \neg E(x,x') \\
\text{encoding}_h(x) &:= \big(\forall x'.\, E(x,x') \rightarrow \text{encoding}_{h-1}(x')\big) \\
&\quad \wedge \big(\forall x'x''.\, E(x,x') \wedge E(x,x'') \wedge x' \not\approx x'' \rightarrow \neg\text{eq}_{h-1}(x',x'')\big) \\
\text{less}_0(x,y) &:= \texttt{false} \\
\text{less}_h(x,y) &:= \exists y'.\, E(y,y') \\
&\qquad \wedge \big(\forall x'.\, E(x,x') \rightarrow \neg\text{eq}_{h-1}(x',y')\big) \\
&\qquad \wedge \big(\forall x''.\, E(x,x'') \wedge \text{less}_{h-1}(y',x'') \rightarrow \exists y''.\, E(y,y'') \wedge \text{eq}_{h-1}(x'',y'')\big) \\
\min(x) &:= \forall x'.\, \neg E(x,x') \\
\text{succ}_0(x,y) &:= \texttt{false} \\
\text{succ}_h(x,y) &:= \exists y'.\, E(y,y') \\
&\qquad \wedge \big(\forall y''.\, E(y,y'') \wedge y'' \not\approx y' \rightarrow \text{less}_{h-1}(y',y'')\big) \\
&\qquad \wedge \big(\forall x'.\, E(x,x') \rightarrow \neg\text{eq}_{h-1}(x',y')\big) \\
&\qquad \wedge \big(\forall y''.\, E(y,y'') \wedge \text{less}_{h-1}(y',y'') \rightarrow \exists x''.\, E(x,x'') \wedge \text{eq}_{h-1}(x'',y'')\big) \\
&\qquad \wedge \big(\forall x''.\, E(x,x'') \wedge \text{less}_{h-1}(y',x'') \rightarrow \exists y''.\, E(y,y'') \wedge \text{eq}_{h-1}(y'',x'')\big) \\
&\qquad \wedge \Big(\neg\min(y') \rightarrow \Big(\big(\exists x'.\, E(x,x') \wedge \min(x')\big) \\
&\qquad\qquad \wedge \big(\forall x'.\, E(x,x') \wedge \text{less}_{h-1}(x',y') \\
&\qquad\qquad\quad \rightarrow \exists z.\, \text{succ}_{h-1}(x',z) \wedge \big(z \approx y' \vee E(x,z)\big)\big)\Big)\Big) \\
\max'_0(x) &:= \forall x'.\, \neg E(x,x') \\
\max'_h(x) &:= \big(\exists y.\, E(x,y) \wedge \min(y)\big) \\
&\quad \wedge \big(\forall x'.\, E(x,x') \rightarrow \max'_{h-1}(x') \vee \big(\exists y.\, E(x,y) \wedge \text{succ}_{h-1}(x',y)\big)\big) \\
\max_0(x) &:= \forall x'.\, \neg E(x,x') \\
\max_h(x) &:= \text{encoding}_h(x) \wedge \max'_h(x)
\end{aligned}
$$

All of these formulas are based on the vocabulary $\langle\{E\},\emptyset\rangle$ and have a length that is polynomial in h. In addition to the formulas defined so far, we use the abbreviation

$$\text{root}(x) := \big(\text{Red}(x) \vee \text{Blue}(x)\big) \wedge \forall x'.\, \neg E(x',x)\ .$$

This brings us one step closer to the definition of the sentences $\varphi_{\mathrm{SF},h}$. In order to illustrate the general ideas, we first construct intermediate sentences. For every $h \geq 0$ we define the sentence φ_h as follows, where we draw some inspiration from the proof of Theorem 2 in [DGKS07a] (Theorem 4.3 in [DGKS07b]):

φ_h

$$\begin{aligned}
\varphi_h := \; & \chi^{\mathrm{r}}_{1,h+4} \wedge \chi^{\mathrm{b}}_{1,h+4} \\
& \wedge \big(\forall x x'.\, \mathrm{Red}(x) \wedge E(x,x') \to \mathrm{Red}(x')\big) \\
& \wedge \big(\forall x x'.\, \mathrm{Blue}(x) \wedge E(x,x') \to \mathrm{Blue}(x')\big) \\
& \wedge \big(\forall x.\, \mathrm{Red}(x) \to \neg\mathrm{Blue}(x)\big) \\
& \wedge \Big(\forall x.\, \big(\neg\mathrm{Red}(x) \wedge \neg\mathrm{Blue}(x)\big) \leftrightarrow \bigvee_{i=1}^{h+3} \big(L^{\mathrm{r}}_i(x) \vee L^{\mathrm{b}}_i(x)\big)\Big) \\
& \wedge \big(\exists y.\, \mathrm{root}(y) \wedge \min(y)\big) \\
& \wedge \Big(\forall x.\, \mathrm{root}(x) \to \max_{h+2}(x) \vee \big(\exists y.\, \mathrm{root}(y) \wedge \mathrm{succ}_{h+2}(x,y)\big)\Big)\,.
\end{aligned}$$

By virtue of Lemma 3.3.16, we conclude that $\mathcal{F}_{h,k} \models \chi^{\mathrm{r}}_{1,h+4} \wedge \chi^{\mathrm{b}}_{1,h+4}$ holds for every $\mathcal{F}_{h,k} \in \mathcal{CF}_h$. Hence, the definition of the class $\mathcal{CF}_h$ entails that every $\mathcal{F}_{h,k} \in \mathcal{CF}_h$ is a model of φ_h.

Obviously, the sentence φ_h is not in SF. The problematic parts are the two last conjuncts. In order to fix this, we need to find separated variants of the formulas $\mathrm{eq}_h(x)$, $\mathrm{encoding}_h(x)$, $\mathrm{less}_h(x,y)$, $\min(x)$, $\max_h(x)$, $\mathrm{succ}_h(x)$, and $\mathrm{root}(x)$. To this end, we define the following color-guarded variants of these formulas: $\mathrm{eq}^{\mathrm{b}}_h(x)$, $\mathrm{eq}^{\mathrm{r}}_h(x)$, $\mathrm{encoding}^{\mathrm{b}}_h(x)$, $\mathrm{encoding}^{\mathrm{r}}_h(x)$, $\mathrm{less}^{\mathrm{b,b}}_h(x,y)$, $\mathrm{less}^{\mathrm{b,r}}_h(x,y)$, $\mathrm{less}^{\mathrm{r,b}}_h(x,y)$, $\mathrm{less}^{\mathrm{r,r}}_h(x,y)$, $\mathrm{root}^{\mathrm{b}}(x)$, $\mathrm{root}^{\mathrm{r}}(x)$, and so on. For the two-argument formulas the superscript *r,b* and similar ones indicate the expected color of the first and second argument, respectively. We exemplary show only the definition of some of these variants, the others are constructed in an analogous way. The definitions are tentative in the sense that we still need to transform the formulas so that they can ultimately be used to construct an SF sentence.

$$\begin{aligned}
\mathrm{eq}^{\mathrm{b,r}}_0(x,y) := \; & \mathrm{Blue}(x) \wedge \mathrm{Red}(y) \\
\mathrm{eq}^{\mathrm{b,r}}_h(x,y) := \; & \mathrm{Blue}(x) \wedge \mathrm{Red}(y) \\
& \wedge \big(\big(\exists u.\, \mathrm{Blue}(u) \wedge E(x,u)\big) \leftrightarrow \big(\exists v.\, \mathrm{Red}(v) \wedge E(y,v)\big)\big) \\
& \wedge \Big(\forall w.\, \mathrm{Blue}(w) \wedge E(x,w) \to \Big(\exists z.\, \mathrm{Red}(z) \wedge E(y,z) \wedge \Big(\forall z'.\, \mathrm{Red}(z') \wedge E(y,z') \\
& \quad \to \Big(\exists w'.\, \mathrm{Blue}(w') \wedge E(x,w') \;\wedge \Big(\forall u.\, \mathrm{Blue}(u) \to \Big(\forall v.\, \mathrm{Red}(v) \\
& \quad\quad \to \Big(\big(u \approx w \wedge v \approx z\big) \vee \big(u \approx w' \wedge v \approx z'\big) \to \mathrm{eq}^{\mathrm{b,r}}_{h-1}(u,v)\Big)\Big)\Big)\Big)\Big)\Big)\Big) \\
\mathrm{encoding}^{\mathrm{b}}_0(x) := \; & \mathrm{Blue}(x) \wedge \forall x'.\, \mathrm{Blue}(x') \to \neg E(x,x') \\
\mathrm{encoding}^{\mathrm{b}}_h(x) := \; & \mathrm{Blue}(x) \\
& \wedge \big(\forall x'.\, \mathrm{Blue}(x') \wedge E(x,x') \to \mathrm{encoding}^{\mathrm{b}}_{h-1}(x')\big) \\
& \wedge \big(\forall x' x''.\, \mathrm{Blue}(x') \wedge \mathrm{Blue}(x'') \wedge E(x,x') \wedge E(x,x'') \wedge x' \not\approx x'' \to \neg\mathrm{eq}^{\mathrm{b,b}}_{h-1}(x',x'')\big) \\
\mathrm{less}^{\mathrm{b,r}}_0(x,y) := \; & \texttt{false} \\
\mathrm{less}^{\mathrm{b,r}}_h(x,y) := \; & \mathrm{Blue}(x) \wedge \mathrm{Red}(y) \\
& \wedge \exists y'.\, \mathrm{Red}(y') \wedge E(y,y') \\
& \quad \wedge \big(\forall x'.\, \mathrm{Blue}(x') \wedge E(x,x') \to \neg\mathrm{eq}^{\mathrm{b,r}}_{h-1}(x',y')\big) \\
& \quad \wedge \big(\forall x''.\, \mathrm{Blue}(x'') \wedge E(x,x'') \wedge \mathrm{less}^{\mathrm{r,b}}_{h-1}(y',x'') \\
& \quad\quad \to \exists y''.\, \mathrm{Red}(y'') \wedge E(y,y'') \wedge \mathrm{eq}^{\mathrm{b,r}}_{h-1}(x'',y'')\big) \\
\min{}^{\mathrm{b}}(x) := \; & \mathrm{Blue}(x) \wedge \forall x'.\, \mathrm{Blue}(x') \to \neg E(x,x')
\end{aligned}$$

$$
\begin{aligned}
&\mathrm{succ}_0^{\mathrm{b,b}}(x,y) := \texttt{false}\\
&\mathrm{succ}_h^{\mathrm{b,b}}(x,y) :=\\
&\quad \mathrm{Blue}(x) \wedge \mathrm{Blue}(y)\\
&\quad \wedge \exists y'.\, \mathrm{Blue}(y') \wedge E(y,y')\\
&\qquad \wedge \big(\forall y''.\, \mathrm{Blue}(y'') \wedge E(y,y'') \wedge y'' \not\approx y' \rightarrow \mathrm{less}_{h-1}^{\mathrm{b,b}}(y',y'')\big)\\
&\qquad \wedge \big(\forall x'.\, \mathrm{Blue}(x') \wedge E(x,x') \rightarrow \neg\mathrm{eq}_{h-1}^{\mathrm{b,b}}(x',y')\big)\\
&\qquad \wedge \big(\forall y''.\, \mathrm{Blue}(y'') \wedge E(y,y'') \wedge \mathrm{less}_{h-1}^{\mathrm{b,b}}(y',y'') \rightarrow \exists x''.\, \mathrm{Blue}(x'') \wedge E(x,x'') \wedge \mathrm{eq}_{h-1}^{\mathrm{b,b}}(x'',y'')\big)\\
&\qquad \wedge \big(\forall x''.\, \mathrm{Blue}(x'') \wedge E(x,x'') \wedge \mathrm{less}_{h-1}^{\mathrm{b,b}}(y',x'') \rightarrow \exists y''.\, \mathrm{Blue}(y'') \wedge E(y,y'') \wedge \mathrm{eq}_{h-1}^{\mathrm{b,b}}(y'',x'')\big)\\
&\qquad \wedge \Big(\neg\mathrm{min}^{\mathrm{b}}(y') \rightarrow \Big(\big(\exists x'.\, \mathrm{Blue}(x') \wedge E(x,x') \wedge \mathrm{min}^{\mathrm{b}}(x')\big)\\
&\qquad\qquad \wedge \Big(\forall x'.\, \mathrm{Blue}(x') \wedge E(x,x') \wedge \mathrm{less}_{h-1}^{\mathrm{b,b}}(x',y')\\
&\qquad\qquad\quad \rightarrow \big(\exists z.\, \mathrm{Blue}(z) \wedge \mathrm{succ}_{h-1}^{\mathrm{b,b}}(x',z) \wedge \big(z \approx y' \vee E(x,z)\big)\big)\Big)\Big)\Big)\\
&\mathrm{succ}_0^{\mathrm{b,r}}(x,y) := \texttt{false}\\
&\mathrm{succ}_h^{\mathrm{b,r}}(x,y) :=\\
&\quad \mathrm{Blue}(x) \wedge \mathrm{Red}(y)\\
&\quad \wedge \exists y'.\, \mathrm{Red}(y') \wedge E(y,y')\\
&\qquad \wedge \big(\forall y''.\, \mathrm{Red}(y'') \wedge E(y,y'') \wedge y'' \not\approx y' \rightarrow \mathrm{less}_{h-1}^{\mathrm{r,r}}(y',y'')\big)\\
&\qquad \wedge \big(\forall x'.\, \mathrm{Blue}(x') \wedge E(x,x') \rightarrow \neg\mathrm{eq}_{h-1}^{\mathrm{b,r}}(x',y')\big)\\
&\qquad \wedge \big(\forall y''.\, \mathrm{Red}(y'') \wedge E(y,y'') \wedge \mathrm{less}_{h-1}^{\mathrm{r,r}}(y',y'') \rightarrow \exists x''.\, \mathrm{Blue}(x'') \wedge E(x,x'') \wedge \mathrm{eq}_{h-1}^{\mathrm{b,r}}(x'',y'')\big)\\
&\qquad \wedge \big(\forall x''.\, \mathrm{Blue}(x'') \wedge E(x,x'') \wedge \mathrm{less}_{h-1}^{\mathrm{r,b}}(y',x'') \rightarrow \exists y''.\, \mathrm{Red}(y'') \wedge E(y,y'') \wedge \mathrm{eq}_{h-1}^{\mathrm{r,b}}(y'',x'')\big)\\
&\qquad \wedge \Big(\neg\mathrm{min}^{\mathrm{r}}(y') \rightarrow \Big(\big(\exists x'.\, \mathrm{Blue}(x') \wedge E(x,x') \wedge \mathrm{min}^{\mathrm{b}}(x')\big)\\
&\qquad\qquad \wedge \Big(\forall x'.\, \mathrm{Blue}(x') \wedge E(x,x') \wedge \mathrm{less}_{h-1}^{\mathrm{b,r}}(x',y')\\
&\qquad\qquad\quad \rightarrow \big(\big(\exists z.\, \mathrm{Blue}(z) \wedge \mathrm{succ}_{h-1}^{\mathrm{b,b}}(x',z) \wedge E(x,z)\big)\\
&\qquad\qquad\qquad \vee \big(\exists z.\, \mathrm{Red}(z) \wedge \mathrm{succ}_{h-1}^{\mathrm{b,r}}(x',z) \wedge z \approx y'\big)\big)\Big)\Big)\Big)\\
&\quad \mathrm{max}'^{\mathrm{b}}_0(x) :=\ \mathrm{Blue}(x) \wedge \forall x'.\, \mathrm{Blue}(x') \rightarrow \neg E(x,x')\\
&\quad \mathrm{max}'^{\mathrm{b}}_h(x) :=\\
&\qquad \mathrm{Blue}(x) \wedge \big(\exists y.\, \mathrm{Blue}(y) \wedge E(x,y) \wedge \mathrm{min}^{\mathrm{b}}(y)\big)\\
&\qquad \wedge \big(\forall x'.\, \mathrm{Blue}(x') \wedge E(x,x') \rightarrow \mathrm{max}'^{\mathrm{b}}_{h-1}(x') \vee \big(\exists y.\, \mathrm{Blue}(y) \wedge E(x,y) \wedge \mathrm{succ}_{h-1}^{\mathrm{b,b}}(x',y)\big)\big)\\
&\quad \mathrm{max}_0^{\mathrm{b}}(x) :=\ \mathrm{Blue}(x) \wedge \forall x'.\, \mathrm{Blue}(x') \rightarrow \neg E(x,x')\\
&\quad \mathrm{max}_h^{\mathrm{b}}(x) :=\ \mathrm{Blue}(x) \wedge \mathrm{encoding}_h^{\mathrm{b}}(x) \wedge \mathrm{max}'^{\mathrm{b}}_h(x)\\
&\quad \mathrm{root}^{\mathrm{b}}(x) :=\ \mathrm{Blue}(x) \wedge \forall x'.\, \mathrm{Blue}(x') \rightarrow \neg E(x',x)
\end{aligned}
$$

Notice that all quantifiers in the above formulas are guarded by *color guards* such that every subformula of the form $\forall x.\, \eta$ for any variable x and any formula η is either equivalent to $\forall x.\, \mathrm{Blue}(x) \rightarrow \eta'$ or to $\forall x.\, \mathrm{Red}(x) \rightarrow \eta'$ for some η'. Similarly, every subformula of the form $\exists y.\, \eta$ for any variable y and any η is either equivalent to $\exists y.\, \mathrm{Blue}(y) \wedge \eta'$ or to $\exists y.\, \mathrm{Red}(y) \wedge \eta'$ for some η'. Regarding the length of formulas, it is easy to check that the length is polynomial in h for most of the formulas. For $\mathrm{succ}_h^{\mathrm{b,b}}(x,y)$ this also holds true, as there is only one recursive reference to $\mathrm{succ}_{h-1}^{\mathrm{b,b}}(x',z)$. For $\mathrm{succ}_h^{\mathrm{b,r}}(x,y)$ the matter is slightly more complicated. We have one recursive reference to $\mathrm{succ}_{h-1}^{\mathrm{b,b}}(x',z)$ and one to $\mathrm{succ}_{h-1}^{\mathrm{b,r}}(x',z)$. But we have already discussed that $\mathrm{succ}_{h-1}^{\mathrm{b,b}}(x',z)$ is a

formula of polynomial length in h. Hence, the overall length of $\mathrm{succ}_h^{\mathrm{b,r}}(x,y)$ is also polynomial in h. Notice that the slightly optimized definition of $\mathrm{succ}_h^{\mathrm{b,b}}(x,y)$ — and, by analogy, also of $\mathrm{succ}_h^{\mathrm{r,r}}(x,y)$ — is necessary to avoid an exponential length of the formulas $\mathrm{succ}_h^{\mathrm{b,r}}(x,y)$ and $\mathrm{succ}_h^{\mathrm{r,b}}(x,y)$.

For the rest of the present proof we use the abbreviations $u \mathrel{\hat{\approx}}^b_{h+4} v$ and $u \mathrel{\hat{\approx}}^r_{h+4} v$ instead of $u \mathrel{\hat{\approx}}^{b,h+4}_{1,h+4} v$ and $u \mathrel{\hat{\approx}}^{r,h+4}_{1,h+4} v$, respectively. We finalize the definition of the above formulas in two steps. In the first step, we push negation signs into the scope of quantifiers so that no quantifier lies within the scope of any negation sign. In the second step, we successively replace every subformula of the form $\exists y.\, \mathrm{Blue}(y) \wedge \eta$ with $\exists y.\, \mathrm{Blue}(y) \wedge \forall v.\, \big(\mathrm{Blue}(v) \wedge y \mathrel{\hat{\approx}}^b_{h+4} v\big) \to \eta\big[y/v\big]$, where we assume v to be fresh (one fresh variable for every replaced subformula). We proceed analogously for every subformula of the form $\exists y.\, \mathrm{Red}(y) \wedge \eta$. The outlined transformations lead to formulas whose length is still polynomial in h.

$u \mathrel{\hat{\approx}}^b_{h+4} v$, $u \mathrel{\hat{\approx}}^r_{h+4} v$

With these tools at hand, we can now define the sentence $\varphi_{\mathrm{SF},h}$ as follows:

$\varphi_{\mathrm{SF},h}$

$$
\begin{aligned}
\varphi_{\mathrm{SF},h} := {} & \chi^{\mathrm{r}}_{1,h+4} \wedge \chi^{\mathrm{b}}_{1,h+4} \\
& \wedge \big(\forall x x'.\, \mathrm{Red}(x) \wedge E(x,x') \to \mathrm{Red}(x')\big) \\
& \wedge \big(\forall x x'.\, \mathrm{Blue}(x) \wedge E(x,x') \to \mathrm{Blue}(x')\big) \\
& \wedge \big(\forall x.\, \mathrm{Red}(x) \to \neg\mathrm{Blue}(x)\big) \\
& \wedge \Big(\forall x.\, \big(\neg\mathrm{Red}(x) \wedge \neg\mathrm{Blue}(x)\big) \leftrightarrow \bigvee_{i=1}^{h+3} \big(L^{\mathrm{r}}_i(x) \vee L^{\mathrm{b}}_i(x)\big)\Big) \\
& \wedge \Big(\big(\exists y.\, \mathrm{Blue}(y) \wedge \mathrm{root}^{\mathrm{b}}(y) \wedge \mathrm{min}^{\mathrm{b}}(y)\big) \vee \big(\exists y.\, \mathrm{Red}(y) \wedge \mathrm{root}^{\mathrm{r}}(y) \wedge \mathrm{min}^{\mathrm{r}}(y)\big)\Big) \\
& \wedge \Big(\forall x.\, \big(\mathrm{Blue}(x) \wedge \mathrm{root}^{\mathrm{b}}(x)\big) \\
& \qquad \to \Big(\mathrm{max}^{\mathrm{b}}_{h+2}(x) \\
& \qquad\quad \vee \big(\exists y.\, \mathrm{Blue}(y) \wedge \big(\forall y'.\, \mathrm{Blue}(y') \wedge y \mathrel{\hat{\approx}}^{\mathrm{b}}_{h+4} y' \to \mathrm{root}^{\mathrm{b}}(y') \wedge \mathrm{succ}^{\mathrm{b,b}}_{h+2}(x,y')\big)\big) \\
& \qquad\quad \vee \big(\exists y.\, \mathrm{Red}(y) \wedge \big(\forall y'.\, \mathrm{Red}(y') \wedge y \mathrel{\hat{\approx}}^{\mathrm{r}}_{h+4} y' \to \mathrm{root}^{\mathrm{r}}(y') \wedge \mathrm{succ}^{\mathrm{b,r}}_{h+2}(x,y')\big)\big)\Big)\Big) \\
& \wedge \Big(\forall x.\, \big(\mathrm{Red}(x) \wedge \mathrm{root}^{\mathrm{r}}(x)\big) \\
& \qquad \to \Big(\mathrm{max}^{\mathrm{r}}_{h+2}(x) \\
& \qquad\quad \vee \big(\exists y.\, \mathrm{Blue}(y) \wedge \big(\forall y'.\, \mathrm{Blue}(y') \wedge y \mathrel{\hat{\approx}}^{\mathrm{b}}_{h+4} y' \to \mathrm{root}^{\mathrm{b}}(y') \wedge \mathrm{succ}^{\mathrm{r,b}}_{h+2}(x,y')\big)\big) \\
& \qquad\quad \vee \big(\exists y.\, \mathrm{Red}(y) \wedge \big(\forall y'.\, \mathrm{Red}(y') \wedge y \mathrel{\hat{\approx}}^{\mathrm{r}}_{h+4} y' \to \mathrm{root}^{\mathrm{r}}(y') \wedge \mathrm{succ}^{\mathrm{r,r}}_{h+2}(x,y')\big)\big)\Big)\Big).
\end{aligned}
$$

The sentence $\varphi_{\mathrm{SF},h}$ is not yet in SF. What still causes trouble are the two subformulas $\mathrm{root}^{\mathrm{b}}(x)$ and $\mathrm{root}^{\mathrm{r}}(x)$ that occur in the antecedents of implications (all other abbreviated subformulas — except for $y \mathrel{\hat{\approx}}^{\mathrm{r}}_{h+4} y'$ and $y \mathrel{\hat{\approx}}^{\mathrm{b}}_{h+4} y'$ — in implications occur exclusively in the respective succedent). Recall that $\mathrm{root}^{\mathrm{b}}(x)$ abbreviates $\mathrm{Blue}(x) \wedge \forall x'.\, \mathrm{Blue}(x') \to \neg E(x,x')$. We define an alternative variant

$$
\mathrm{root'}^{\mathrm{b}}(x) := \mathrm{Blue}(x) \wedge \forall x'.\, \mathrm{Blue}(x') \to \exists y'.\, \mathrm{Blue}(y') \wedge x' \mathrel{\hat{\approx}}^b_{h+4} y' \wedge \neg E(y',x)\ .
$$

Then, a prenex form of any sentence of the form $\forall x.\, \mathrm{Blue}(x) \wedge \mathrm{root'}^{\mathrm{b}}(x) \to \eta$ belongs to SF, if η satisfies the necessary separateness conditions. Shifting the quantifiers from the antecedent of the implication to the front yields

$$
\forall x \exists x' \forall y'.\, \mathrm{Blue}(x) \wedge \Big(\mathrm{Blue}(x) \wedge \big(\mathrm{Blue}(x') \to \mathrm{Blue}(y') \wedge x' \mathrel{\hat{\approx}}^b_{h+4} y' \wedge \neg E(y',x)\big)\Big) \to \eta\ ,
$$

where we can easily check that the separateness conditions are satisfied for the antecedent of the implication. The formula $\mathrm{root'}^{\mathrm{r}}(x)$ is defined in analogy to $\mathrm{root'}^{\mathrm{b}}(x)$. We now replace the

subformulas $\mathrm{root}^{\mathrm{b}}(x)$ and $\mathrm{root}^{\mathrm{r}}(x)$ in $\varphi_{\mathrm{SF},h}$ that occur in the antecedents of implications with the alternatives $\mathrm{root}'^{\mathrm{b}}(x)$ and $\mathrm{root}'^{\mathrm{r}}(x)$. The result is almost an SF sentence, but technically only a variant of $\varphi_{\mathrm{SF},h}$ is in SF in which all quantifiers are shifted to the front. Shifting quantifiers can be done in a straightforward fashion. This finishes the construction of the sentences $\varphi_{\mathrm{SF},h}$, each of which has a length polynomial in h. Like for the sentences φ_h before, it is not hard to check that every $\mathcal{F}_{h,k} \in \mathcal{CF}_h$ is a model of $\varphi_{\mathrm{SF},h}$.

For the rest of this proof we fix some nonnegative integer $h \geq 0$. Consider the sentence $\varphi_{\mathrm{SF},h}$ and suppose there is some equivalent sentence ψ in Gaifman normal that has a length of at most $2^{\uparrow h} - 1$. Let $\eta_1, \ldots, \eta_L$ be a list of all the basic local sentences that occur in ψ. Each η_ℓ has the form

ψ

η_ℓ, L

$$\exists y_1^\ell \ldots y_{j_\ell}^\ell . \bigwedge_{1 \leq i < j \leq j_\ell} \mathrm{dist}_{>2r_\ell}(y_i^\ell, y_j^\ell) \wedge \bigwedge_{1 \leq i \leq j_\ell} \psi_\ell(y_i^\ell) \ .$$

Let $K := \sum_{1 \leq \ell \leq L} j_\ell$. By virtue of our assumption regarding the length of ψ, we know that $L \leq K < 2^{\uparrow h}$.

K

The class $\mathcal{CF}_h$ contains $2^{\uparrow h+2}$ forests $\mathcal{F}_{h,k}$. For every set $S \subseteq [L]$ we define $\mathcal{C}_S$ to be the class of all forests $\mathcal{F}_{h,k} \in \mathcal{CF}_h$ for which we have $\mathcal{F}_{h,k} \models \eta_\ell$ if and only if $\ell \in S$. Then, there must be some $S_* \subseteq [L]$ for which $\mathcal{C}_{S_*}$ contains at least

$\mathcal{C}_S$

S_*

$$\frac{2^{\uparrow h+2}}{2^L} \ > \ \frac{2^{\uparrow h+2}}{2^{\uparrow h+1}} \ \geq \ 2^{\uparrow h} \ > \ L$$

structures. For every $\mathcal{F}_{h,k} \in \mathcal{C}_{S_*}$ and every $\ell \in S_*$ there are nodes $\mathsf{c}_1^{k,\ell}, \ldots, \mathsf{c}_{j_\ell}^{k,\ell}$ such that $\mathcal{F}_{h,k} \models \psi_\ell(\mathsf{c}_i^{k,\ell})$ holds for every i. Let $\mathsf{D}_k := \bigcup_{1 \leq \ell \leq L} \{\mathsf{c}_1^{k,\ell}, \ldots, \mathsf{c}_{j_\ell}^{k,\ell}\}$. We distinguish the following two cases, where the set $\mathsf{F}_k^{\mathrm{r}} := \mathrm{Red}^{\mathcal{F}_{h,k}} \cup \bigcup_{1 \leq j \leq h-2} L_j^{\mathrm{r}\mathcal{F}_{h,k}}$ denotes the *red part of* $\mathcal{F}_{h,k}$:

$\mathsf{c}_i^{k,\ell}$

D_k

$\mathsf{F}_k^{\mathrm{r}}$

If there is any k for which none of the domain elements $\mathsf{c}_i^{k,\ell}$ belongs to $\mathsf{F}_k^{\mathrm{r}}$, then none of the neighborhoods $\mathcal{N}_{\mathcal{F}_{h,k}}^{r_\ell}(\mathsf{c}_i^{k,\ell})$ contains any elements from $\mathsf{F}_k^{\mathrm{r}}$. Hence, the substructure $\mathcal{F}'$ of $\mathcal{F}_{h,k}$ induced by the domain $\mathsf{F}_{h,k} \setminus \mathsf{F}_k^{\mathrm{r}}$ still satisfies any sentence η_ℓ if and only if $\ell \in S$. Consequently, we have $\mathcal{F}' \models \varphi_{\mathrm{SF},h}$. But this contradicts the fact that $\varphi_{\mathrm{SF},h}$ stipulates the presence of all trees $\mathcal{T}(0), \ldots, \mathcal{T}(2^{\uparrow h+2} - 1)$ in $\mathcal{F}'$, in particular the tree $\mathcal{T}(k)$, which does not occur in $\mathcal{F}'$.

$\mathcal{F}'$

Otherwise, the red part $\mathsf{F}_k^{\mathrm{r}}$ of each of the structures $\mathcal{F}_{h,k} \in \mathcal{C}_{S_*}$ contains at least one of the $\mathsf{c}_i^{k,\ell} \in \mathsf{D}_k$. Recall that there are at most $L < 2^{\uparrow h}$ distinct r_ℓ-local formulas in ψ, but at least $2^{\uparrow h}$ structures $\mathcal{F}_{h,k}$ in $\mathcal{C}_{S_*}$. Let $\mathcal{F}_{h,k}^{\mathrm{r}}$ be the substructure of $\mathcal{F}_{h,k}$ induced by its red part $\mathsf{F}_k^{\mathrm{r}}$. Then, because of $L < 2^{\uparrow h} \leq |\mathcal{C}_{S_*}|$, there must be some k_* such that for every ψ_ℓ with $\ell \in S_*$ and every $\mathsf{d} \in \mathsf{F}_{k_*}^{\mathrm{r}}$ with $\mathsf{d} = \mathsf{c}_i^{k_*,\ell}$ for some i there is some $k_\mathsf{d} \neq k_*$ such that $\mathcal{F}_{h,k_\mathsf{d}}^{\mathrm{r}} \models \psi_\ell(\mathsf{e})$ for some $\mathsf{e} \in \{\mathsf{c}_1^{k_\mathsf{d},\ell}, \ldots, \mathsf{c}_{j_\ell}^{k_\mathsf{d},\ell}\}$.

$\mathcal{F}_{h,k}^{\mathrm{r}}$

k_d

We create a new structure $\mathcal{A}$ that is the disjoint union of the following structures:

$\mathcal{A}$

- the substructure $\mathcal{F}_{h,k_*}^{\mathrm{b}}$ of $\mathcal{F}_{h,k_*}$ induced by the *blue part of* $\mathcal{F}_{h,k_*}$, i.e. by the set $\mathrm{Blue}^{\mathcal{F}_{h,k_*}} \cup \bigcup_{1 \leq j \leq h-2} L_j^{\mathrm{b}\mathcal{F}_{h,k_*}}$,
- the substructures $\mathcal{F}_{h,k_\mathsf{d}}^{\mathrm{r}}$ for every $\ell \in S_*$ and every $\mathsf{d} \in \mathsf{F}_{k_*}^{\mathrm{r}}$ with $\mathsf{d} = \mathsf{c}_i^{k_*,\ell}$ for some i.

$\mathcal{F}_{h,k}^{\mathrm{b}}$

First of all, we notice that $\mathcal{A}$ does not contain a representation of the tree $\mathcal{T}(k_*)$. Hence, $\mathcal{A} \not\models \varphi_{\mathrm{SF},h}$.

On the other hand, we have $\mathcal{A} \models \eta_\ell$ for every $\ell \in S_*$. More precisely, for every $\ell \in S_*$ we observe

$$\mathcal{A} \models \bigwedge_{1 \leq i < j \leq j_\ell} \mathrm{dist}_{>2r_\ell}(\mathsf{e}_i^\ell, \mathsf{e}_j^\ell) \wedge \bigwedge_{1 \leq i \leq j_\ell} \psi_\ell(\mathsf{e}_i^\ell)$$

where the domain elements e_i^ℓ are defined as follows. For every i with blue $\mathsf{c}_i^{k_*,\ell}$, i.e. $\mathsf{c}_i^{k_*,\ell} \in \mathsf{D}_{k_*} \setminus \mathsf{F}^{\mathrm{r}}_{k_*}$, we set $\mathsf{e}_i^\ell := \mathsf{c}_i^{k_*,\ell}$. For every i with red $\mathsf{c}_i^{k_*,\ell}$, i.e. $\mathsf{c}_i^{k_*,\ell} \in \mathsf{F}^{\mathrm{r}}_{k_*}$, we set $\mathsf{e}_i^\ell := \mathsf{e}$ for the element e that originates from the substructure $\mathcal{F}^{\mathrm{r}}_{h,k_\mathsf{d}}$ for which $\mathcal{F}^{\mathrm{r}}_{h,k_\mathsf{d}} \models \psi_\ell(\mathsf{e})$. Since the formulas ψ_ℓ are r_ℓ-local and since there are no links in the Gaifman graph of $\mathcal{A}$ between the nodes stemming from the disjoint substructures forming $\mathcal{A}$, we get

$$\mathcal{A} \models \bigwedge_{1 \leq i < j \leq j_\ell} \mathrm{dist}_{>2r_\ell}(\mathsf{e}_i^\ell, \mathsf{e}_j^\ell) \ .$$

Moreover, for every $\ell \in [L] \setminus S_*$ we have $\mathcal{F}_{h,k_*} \models \neg\eta_\ell$ and $\mathcal{F}_{h,k_\mathsf{d}} \models \neg\eta_\ell$ for all $\mathsf{d} \in \mathsf{F}^{\mathrm{r}}_{k_*}$. That is, for any structure $\mathcal{F}$ among these structures we have

$$\mathcal{F} \models \forall y_1^\ell \ldots y_{j_\ell}^\ell . \Big(\bigwedge_{1 \leq i < j \leq j_\ell} \mathrm{dist}_{>2r_\ell}(y_i^\ell, y_j^\ell) \Big) \rightarrow \bigvee_{1 \leq i \leq j_\ell} \neg\psi_\ell(y_i^\ell) \ .$$

Let $\mathcal{F}'$ be the corresponding substructure among $\mathcal{F}^{\mathrm{b}}_{h,k_*}$ and $\mathcal{F}^{\mathrm{r}}_{h,k_\mathsf{d}}$ with $\mathsf{d} \in \mathsf{F}^{\mathrm{r}}_{k_*}$. The Gaifman graph of $\mathcal{F}'$ can be obtained from $\mathcal{F}$'s Gaifman graph by entirely removing one connected component and leaving the rest untouched. Since all formulas ψ_ℓ are r_ℓ-local, $\mathcal{F}'$ also satisfies the above sentence. In other words, we have $\mathcal{F}' \not\models \eta_\ell$. As $\mathcal{A}$ is the disjoint union of all these substructures, this also yields $\mathcal{A} \not\models \eta_\ell$ for every $\ell \in [L] \setminus S_*$.

In summary, for every $\ell \in L$ we have $\mathcal{A} \models \eta_\ell$ if and only if $\ell \in S^*$. Consequently, $\mathcal{A} \models \psi$. This contradicts our earlier observation $\mathcal{A} \not\models \varphi_{\mathrm{SF},h}$ and our assumption that ψ is semantically equivalent to $\varphi_{\mathrm{SF},h}$.

As both cases lead to a contradiction, the sentence ψ cannot exist. □

One of the interesting aspects of the proof of Theorem 3.3.18 is that – in contrast to the proof of Theorem 3.3.11 — we do not restrict the whole domain to a finite set. Instead, we only restrict subdomains whose elements are affected by non-separated quantification. In Section 5.3.1 we will show related techniques with which we can enforce large-sized subdomains in models for SF sentences.

The results in the present section evidently show that the transformation outlined in Lemma 3.3.10 is a useful tool for proving lower bounds. On the one hand, we have derived lower bounds regarding the computational hardness of SF-Sat. On the other hand, we have shown that for every positive k SF sentences can be k-fold exponentially more succinct than equivalent BSR sentences or equivalent sentences in Gaifman normal form. Unfortunately, the presented translation methodology does not help in the quest for new decidable first-order fragments. The reason is simply that we already need arguments leading to a small model property before we can start the translation process, as we need information about the size of the models that have to be considered.

3.4 The Generalized Bernays–Schönfinkel–Ramsey Fragment (GBSR)

In this section we extend the separated fragment even further. Recall that SF contains relational sentences $\exists\bar{z}\forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1 \ldots \forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n . \psi$ in which the sets $\bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_n$ and $\bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_n$ are separated. The exemption of the leading existential quantifier block from the separateness conditions may lead to certain co-occurrences of existentially and universally quantified variables in atoms. Such co-occurrences are in some sense *benign*, as they do not pose an obstacle to the construction of algorithms that decide the satisfiability problem. For the moment it is not clear whether the leading existential quantifier block is the only possible source for such nicely behaving co-occurrences with universal variables. Indeed, we shall see shortly that the benign co-occurrences in SF are only the first sign of a more general notion. Exploring this emerging pattern leads to the definition of another decidable fragment of first-order logic, the *generalized Bernays–Schönfinkel–Ramsey*

fragment — GBSR for short. Although the formal definition of GBSR subsumes all SF sentences, it can be considered a natural generalization of the original Bernays–Schönfinkel–Ramsey fragment; hence the name.

Intuitively speaking, a GBSR sentence φ has the form $\forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\ldots\forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n.\,\psi$ with quantifier-free ψ that may contain equality and possesses the following properties. Each atom in φ only contains variables from a subsequence of φ's quantifier prefix of the form $\exists^*\forall^*$. If two atoms share a universally quantified variable, the same quantifier subsequence is used for both atoms.[3] Notice that the idea of restricting subsequences of nested quantifiers (instead of prefixes of sentences in prenex normal form) has also been used for other fragments, e.g. Maslov's fragment K (cf. page 25).

Definition 3.4.1 (Generalized Bernays–Schönfinkel–Ramsey fragment (GBSR))**.**
Let $\varphi := \forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\ldots\forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n.\,\psi$ be any relational first-order sentence with equality and quantifier-free ψ. Let At *be the set of all atoms occurring in φ and let $\bar{\mathrm{x}} := \bar{\mathrm{x}}_1\cup\ldots\cup\bar{\mathrm{x}}_n$ and $\bar{\mathrm{y}} := \bar{\mathrm{y}}_1\cup\ldots\cup\bar{\mathrm{y}}_n$. The sentence φ belongs to the* generalized Bernays–Schönfinkel–Ramsey fragment (GBSR) *if and only if we can partition* At *into sets $\mathrm{At}_0, \mathrm{At}_1, \ldots, \mathrm{At}_n$ such that*
(i) for every i, $0 \le i \le n$, we have $\mathrm{vars}(\mathrm{At}_i) \subseteq \bar{\mathrm{y}}_1\cup\ldots\cup\bar{\mathrm{y}}_i\cup\bar{\mathrm{x}}_{i+1}\cup\ldots\cup\bar{\mathrm{x}}_n$, and
(ii) for all distinct i, j, $0 \le i < j \le n$, we have $\mathrm{vars}(\mathrm{At}_i)\cap\mathrm{vars}(\mathrm{At}_j)\cap\bar{\mathrm{x}} = \emptyset$.

Clearly, the main difference between SF and GBSR lies in the concession policy regarding benign co-occurrences of existential and universal variables. The following example gives a first impression of GBSR sentences and how they can be translated into BSR.

Example 3.4.2. *Consider the first-order sentence $\varphi := \exists u\forall x\exists y\forall z.\,\big(P(u,z)\wedge Q(u,x)\big)\vee\big(P(y,z)\wedge Q(u,y)\big)$. It belongs to GBSR, as witnessed by the following partition of its atoms: $\mathrm{At}_0 = \emptyset$, $\mathrm{At}_1 = \{Q(u,x)\}$, $\mathrm{At}_2 = \{P(u,z), P(y,z), Q(u,y)\}$, $\mathrm{At}_3 = \emptyset$. Obviously, φ neither belongs to BSR nor to SF. As universal quantification does not distribute over disjunction, the quantifier $\forall z$ cannot be shifted inwards with the standard quantifier shifting rules from Lemma 1.0.1 alone. However, it turns out that the transformation methods that we have first met in Section 2 and which we applied to transform SF sentences into equivalent BSR sentences also facilitate a translations of GBSR sentences into BSR sentences. We shall elaborate on this in Section 3.5. For φ we get the equivalent BSR sentence*

$$\begin{aligned}\varphi' := \exists uy\forall xzv.\,&\big(\big(P(u,x)\vee P(y,x)\big)\wedge P(u,x)\wedge Q(u,x)\big)\\ &\vee\big(\big(P(u,z)\vee P(y,z)\big)\wedge Q(u,y)\wedge Q(u,z)\big)\\ &\vee\big(\big(P(u,v)\vee P(y,v)\big)\wedge Q(u,y)\wedge P(y,v)\big)\ .\end{aligned}$$

In contrast to SF, it is not immediately clear whether membership in GBSR can be tested efficiently. However, we can easily show that this is indeed the case.

Theorem 3.4.3. *Deciding membership of first-order sentences in GBSR can be done deterministically in time that is polynomial in the length of any reasonable encoding of the input sentence.*

Proof sketch. Suppose we are given a first-order sentence $\varphi := \forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\ldots\forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n.\,\psi$ with quantifier-free ψ. Let $\bar{\mathrm{x}} := \bar{\mathrm{x}}_1\cup\ldots\cup\bar{\mathrm{x}}_n$ and $\bar{\mathrm{y}} := \bar{\mathrm{y}}_1\cup\ldots\cup\bar{\mathrm{y}}_n$. We define the undirected graph $\mathcal{G}_\varphi := \langle V, E\rangle$ by setting $V := \bar{\mathrm{x}}$ and $E := \{\langle x, x'\rangle \mid$ there is an atom in φ containing both x and $x'\}$. A *connected component in $\mathcal{G}_\varphi$* is a maximal subset $C \subseteq V$ such that for all distinct variables $x, x' \in C$ the transitive closure of E contains the pair $\langle x, x'\rangle$. The set of all connected components in $\mathcal{G}_\varphi$ forms a partition of V. For every connected component C in $\mathcal{G}_\varphi$ we denote by $\mathrm{at}(C)$ the *set of all atoms in φ which contain at least one variable from C*. For every index k, $1 \le k \le n$, we denote by at_k the smallest set of atoms such that at_k contains all atoms taken from φ in which variables from $\bar{\mathrm{y}}_k$ occur and for every connected component C in $\mathcal{G}_\varphi$ containing a variable $x \in \mathrm{vars}(\mathrm{at}_k)$ we have $\mathrm{at}(C) \subseteq \mathrm{at}_k$.

We partition the set of all atoms in φ into parts $\widetilde{\mathrm{at}}_0, \widetilde{\mathrm{at}}_1, \ldots, \widetilde{\mathrm{at}}_n$, where $\widetilde{\mathrm{at}}_k := \mathrm{at}_k \setminus \bigcup_{\ell>k}\mathrm{at}_\ell$ and $\widetilde{\mathrm{at}}_0$ collects all atoms in φ that do not belong to any $\widetilde{\mathrm{at}}_k$ with $k > 0$. For every k, $0 \le k \le n$, we write X_k to address the set $\mathrm{vars}(\widetilde{\mathrm{at}}_k)\cap\bar{\mathrm{x}}$.

[3] This intuitive explanation is based on the report of an anonymous referee the author of the present thesis received for a submission to LICS 2018.

Claim I:

(i) For all distinct indices k, ℓ we have $X_k \cap X_\ell = \emptyset$.

(ii) For every k we have $\mathrm{vars}(\widetilde{\mathrm{at}}_k) \cap \bar{\mathrm{y}} \subseteq \bigcup_{1 \leq \ell \leq k} \bar{\mathrm{y}}_\ell$.

(iii) If we have $\mathrm{vars}(\mathrm{at}_k) \cap \bar{\mathrm{x}}_\ell = \emptyset$ for all k, ℓ with $1 \leq \ell \leq k \leq n$, then we have for every k', $1 \leq k' \leq n$, that $X_{k'} \subseteq \bigcup_{k' < \ell' \leq n} \bar{\mathrm{x}}_{\ell'}$.

Proof:

Ad (i): Suppose there are distinct indices k, ℓ, $k < \ell$, and a variable $x \in X_k \cap X_\ell$. Then, there must be atoms $A_k \in \widetilde{\mathrm{at}}_k \subseteq \mathrm{at}_k$ and $A_\ell \in \widetilde{\mathrm{at}}_\ell \subseteq \mathrm{at}_\ell$, both containing x. Let C denote the (unique) connected component in $\mathcal{G}_\varphi$ to which x belongs. By definition of $\mathrm{at}(C)$, both A_k and A_ℓ belong to $\mathrm{at}(C)$. Therefore, we have $\{A_k, A_\ell\} \subseteq \mathrm{at}(C) \subseteq \mathrm{at}_\ell$. But since $\widetilde{\mathrm{at}}_k \subseteq \mathrm{at}_k \setminus \mathrm{at}_\ell$, A_k cannot belong to at_k. This yields a contradiction.

Ad (ii): Let $k \leq n$ be some non-negative integer. Since for any $\ell > k$ at_ℓ contains all atoms in which a variable $y \in \bar{\mathrm{y}}_\ell$ occurs, $\widetilde{\mathrm{at}}_k \subseteq \mathrm{at}_k \setminus \mathrm{at}_\ell$ cannot contain any occurrence of y.

Ad (iii): Let $k \leq n$ be some non-negative integer. Suppose we have $\mathrm{vars}(\mathrm{at}_k) \cap \bigcup_{\ell \leq k} \bar{\mathrm{x}}_\ell = \emptyset$. Because of $X_k = \mathrm{vars}(\widetilde{\mathrm{at}}_k) \cap \bar{\mathrm{x}} \subseteq \mathrm{vars}(\mathrm{at}_k) \cap \bar{\mathrm{x}}$, we conclude $X_k \cap \bigcup_{\ell' \leq k} \bar{\mathrm{x}}_{\ell'} = \emptyset$. Hence, we have $X_k \subseteq \bigcup_{\ell > k} \bar{\mathrm{x}}_\ell$. ◊

Claim II: The sentence φ belongs to *GBSR* if and only if for all k, ℓ with $1 \leq \ell \leq k \leq n$ we have $\mathrm{vars}(\mathrm{at}_k) \cap \bar{\mathrm{x}}_\ell = \emptyset$.

Proof: The *if*-direction follows immediately from Claim I, if we set $\mathrm{At}_k := \widetilde{\mathrm{at}}_k$ for every k.

The *only if*-direction can be argued as follows. For every i, $0 \leq i \leq n$, let $X'_i := \mathrm{vars}(\mathrm{At}_i) \cap \bar{\mathrm{x}}$. Consider the graph $\mathcal{G}_\varphi$. Since the $X'_1, \ldots, X'_n$ are pairwise disjoint, they induce subgraphs of $\mathcal{G}_\varphi$ that are not connected to one another. Moreover, for every connected component C in $\mathcal{G}_\varphi$ there is one X'_i such that $C \subseteq X'_i$. This entails that for every connected component C in $\mathcal{G}_\varphi$ there is some At_i such that $\mathrm{at}(C) \subseteq \mathrm{At}_i$. By definition of at_k, we have $\mathrm{at}_k \subseteq \mathrm{At}_k \cup \ldots \cup \mathrm{At}_n$ and, moreover, $\mathrm{vars}(\mathrm{at}_k) \cap \bar{\mathrm{x}} \subseteq \bar{\mathrm{x}}_{k+1} \cup \ldots \cup \bar{\mathrm{x}}_n$. This means, $\mathrm{vars}(\mathrm{at}_k) \cap \bar{\mathrm{x}}_\ell = \emptyset$ for every ℓ, $1 \leq \ell \leq k$. ◊

Claim II yields a criterion to decide whether φ belongs to GBSR or not. It remains to convince ourselves that this criterion can be checked deterministically in polynomial time. Given φ, the graph $\mathcal{G}_\varphi = \langle V, E \rangle$ can be constructed in time that is quadratic in $\|\varphi\|$. We observe $|V| = |\bar{\mathrm{x}}| \leq |\mathrm{len}(\varphi)|$ and $|E| \leq \binom{|\bar{\mathrm{x}}|}{2} \leq \big(\mathrm{len}(\varphi)\big)^2$. Using efficient disjoint-set data structures, the connected components of $\mathcal{G}_\varphi$ and the sets at_k can be computed in time that is polynomial in $\|\varphi\|$. The sum of the lengths of the atoms in $\mathrm{at}_1, \ldots, \mathrm{at}_n$ is at most $n \cdot \mathrm{len}(\varphi) \leq \big(\mathrm{len}(\varphi)\big)^2$. Finally, the test whether we have $\mathrm{vars}(\mathrm{at}_k) \cap \bar{\mathrm{x}}_\ell = \emptyset$ for all k, ℓ with $1 \leq \ell \leq k \leq n$ can be done in time that is polynomial in $\|\varphi\|$. □

GBSR has been advertised as an extension of SF, which in turn contains BSR and MFO. Indeed, given an SF sentence $\chi := \exists \bar{\mathrm{z}} \forall \bar{\mathrm{u}}_1 \exists \bar{\mathrm{v}}_1 \ldots \forall \bar{\mathrm{u}}_n \exists \bar{\mathrm{v}}_n . \chi'$, we can partition the set of χ's atoms into two nonempty sets $\mathrm{At}_1, \mathrm{At}_n$ such that $\mathrm{vars}(\mathrm{At}_1) \subseteq \bar{\mathrm{z}} \cup \bar{\mathrm{u}}_1 \cup \ldots \cup \bar{\mathrm{u}}_n$ and $\mathrm{vars}(\mathrm{At}_n) \subseteq \bar{\mathrm{z}} \cup \bar{\mathrm{v}}_1 \cup \ldots \cup \bar{\mathrm{v}}_n$. This partition obviously satisfies the requirements of Definition 3.4.1. On the other hand, the sentence φ from Example 3.4.2 belongs to GBSR but not to SF. Hence, GBSR is a proper extension of SF.

Proposition 3.4.4. *GBSR properly contains SF and, hence, BSR and MFO.*

By Theorem 3.1.5, GBSR in addition semantically subsumes $\mathrm{MFO}_{\approx}$, like SF does.

We shall discuss two ways of showing that the satisfiability problem for GBSR sentences (*GBSR-Sat*) is decidable. The first approach is of a syntactic nature, based on an effective translation from GBSR into BSR. We elaborate on this in the next section. The second approach uses model-theoretic techniques to directly establish a small model property. In Section 4.2 we will consider model-checking games for GBSR sentences and prove the existence of a special kind of

GBSR-Sat

winning strategies that induce finite models for satisfiable GBSR sentences. Both approaches in the end lead to upper bounds on the computational complexity of GBSR-Sat. We shall pick up on this topic in Chapter 5.

3.5 Translation of GBSR into BSR

Like for SF there is an effective equivalence-preserving translation from GBSR into BSR. It essentially follows the same lines as the SF-BSR translation and is also mainly based on the standard laws of Boolean algebra and quantifier shifting. However, the additional benign co-occurrences of universally and existentially quantified variables in GBSR sentences require a bit more attention. Roughly speaking, we iteratively (re-)transform a given GBSR sentence into particular syntactic shapes and apply quantifier shifting so that we eventually obtain a formula in which no existential quantifier occurs within the scope of any universal quantifier. We then shift all quantifiers outwards again — existential quantifiers first —, renaming bound variables as necessary. The final result is a BSR sentence. Since GBSR contains SF, Theorem 3.2.7 entails that there is no elementary upper bound on the blowup that we incur in any equivalence-preserving translation from GBSR into BSR. On the other hand, the blowup for GBSR-BSR translations will not be significantly worse than in the case of SF-BSR translations. It seems that in this sense GBSR does not offer much more succintness compared to BSR when describing first-order properties than SF does.

The accuracy of our analysis of the translation from SF into BSR benefited from measuring the degree of separateness between existentially quantified variables that stem from distinct quantifier blocks. Any attempt to a similar analysis for the translation from GBSR into BSR requires a similar, yet more sophisticated measure. The key difference is that we have to deal with the additional benign co-occurrences of universally and existentially quantified variables. An appropriate measure for GBSR sentences is the following

Definition 3.5.1 (Degree of interaction for GBSR sentences)**.** *Consider any GBSR sentence* $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \psi$ *in which* ψ *is quantifier free. In analogy to Definition 3.2.1, we say that any variable* $y \in \bar{\mathrm{y}}_j$ has index j*, denoted* $\mathrm{idx}(y) = j$*. For any nonempty set* $V \subseteq \mathrm{vars}(\varphi)$ *and any positive integer* k *we say that* V has degree k in φ*, denoted* $\partial(V, \varphi) = k$*, if* k *is the maximal number of distinct variables* $v_1, \ldots, v_k \in V$ *with* $\mathrm{idx}(v_1) < \ldots < \mathrm{idx}(v_k)$*. We say that the GBSR sentence* φ has degree k*, denoted* $\partial_{\exists\forall}(\varphi) = k$*, if* k *is the smallest positive integer such that we can partition* $\mathrm{vars}(\varphi)$ *into* $m > 0$ *parts* $V_1, \ldots, V_m$ *that are all pairwise separated in* φ *and for which* $k = \max\{k_j \mid \partial(V_j, \varphi) = k_j, 1 \leq j \leq m\}$.

idx(y)

$\partial_{\exists\forall}(\varphi)$

The analysis of the incurred blowup when translating GBSR into BSR will be significantly more accurate when we base it on the degree of interaction of variables from distinct quantifier blocks rather than on the number of occurring $\forall\exists$-alternations, for instance. Even this difference in the outcome of the analysis cannot be elementarily bounded in the worst case.

Lemma 3.5.2. *Let* $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \psi$ *be any GBSR sentence with quantifier-free* ψ*. There exists a quantifier-free first-order formula* $\psi'(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ *such that* $\varphi' := \exists \bar{\mathrm{u}} \forall \bar{\mathrm{v}} . \psi'(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ *is in standard form and equivalent to* φ*, and all literals in* φ' *also occur in* φ *(modulo variable renaming). Moreover,* $\bar{\mathrm{u}}$ *contains at most* $|\bar{\mathrm{y}}|^2 \cdot \partial_{\exists\forall}(\varphi) \cdot \big(2^{\uparrow \partial_{\exists\forall}(\varphi)}(2 \cdot \mathrm{len}(\varphi))\big)^{\partial_{\exists\forall}(\varphi)}$ *leading existential quantifiers.*

Proof sketch. Without losing generality, we assume that φ is in standard form. Let At, $\bar{\mathrm{x}}$, and $\bar{\mathrm{y}}$ be defined as in Definition 3.4.1 and let $\mathrm{At}_0, \mathrm{At}_1, \ldots, \mathrm{At}_n$ be some partition in accordance with Definition 3.4.1. Let $\mathcal{L}_0, \mathcal{L}_1, \ldots, \mathcal{L}_n$ be the corresponding partition of the set of literals occurring in φ, i.e. every At_i is exactly the set of atoms occurring in $\mathcal{L}_i$. Furthermore, let $X_i := \mathrm{vars}(\mathcal{L}_i) \cap \bar{\mathrm{x}}$. By Definition 3.4.1, we observe the following:

At_i, $\mathcal{L}_i$, X_i

(I) For all distinct indices k, ℓ we have $X_k \cap X_\ell = \emptyset$.

(II) For every k we have $\mathrm{vars}(\mathcal{L}_k) \subseteq \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_k \cup \bar{\mathrm{x}}_{k+1} \cup \ldots \cup \bar{\mathrm{x}}_n$.

We transform φ into an equivalent formula in CNF of the form

$$\forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\ldots\forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n.\bigwedge_{i\in I}\chi^{(1)}_{i,0}(\bar{\mathrm{x}}_1,\ldots,\bar{\mathrm{x}}_n)\vee\chi^{(1)}_{i,1}(\bar{\mathrm{y}}_1,\bar{\mathrm{x}}_2,\ldots,\bar{\mathrm{x}}_n)\vee\ldots \vee\chi^{(1)}_{i,n-1}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_{n-1},\bar{\mathrm{x}}_n)\vee\chi^{(1)}_{i,n}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_n)$$

where

(a) the set I is a finite set of indices,

(b) the $\chi^{(1)}_{i,n}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_n)$ are disjunctions of literals $\bigvee_{k\in K_i}L_k(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_n)$ where the sets K_i are finite, pairwise disjoint — also disjoint with I — sets of indices,

(c) the $\chi^{(1)}_{i,j}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_j,\bar{\mathrm{x}}_{j+1},\ldots,\bar{\mathrm{x}}_n)$ contain exclusively literals from $\mathcal{L}_j$.

By virtue of Lemma 3.2.4, there is an equivalent formula of the form

$$\varphi' := \forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\ldots\forall\bar{\mathrm{x}}_n.\bigwedge_{S\in\mathcal{P}I\setminus\emptyset}\Big(\bigvee_{i\in S}\bigvee_{j=0}^{n-1}\chi^{(1)}_{i,j}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_j,\bar{\mathrm{x}}_{j+1},\ldots,\bar{\mathrm{x}}_n)\Big) \vee\bigvee_{f\in\mathcal{F}}\Big(\exists\bar{\mathrm{y}}_n.\bigwedge_{i\in S}L_{f(i)}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_n)\Big)$$

where $\mathcal{F}$ is the set of all selection functions over the family of index sets $(K_i)_{i\in I}$. Applying ordinary quantifier shifting and exploiting the disjointness of the sets X_j (cf. (I)), we shift the universal quantifier block $\forall\bar{\mathrm{x}}_n$ inwards and thus obtain

$$\varphi'' := \forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\ldots\exists\bar{\mathrm{y}}_{n-1}.\bigwedge_{S\in\mathcal{P}I\setminus\emptyset}\Big(\Big(\bigvee_{j=0}^{n-1}\forall(\bar{\mathrm{x}}_n\cap X_j).\bigvee_{i\in S}\chi^{(1)}_{i,j}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_j,\bar{\mathrm{x}}_{j+1},\ldots,\bar{\mathrm{x}}_n)\Big) \vee\forall(\bar{\mathrm{x}}_n\cap X_n).\bigvee_{f\in\mathcal{F}}\exists\bar{\mathrm{y}}_n.\bigwedge_{i\in S}L_{f(i)}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_n)\Big)\,.$$

Notice that any distinct $\chi^{(1)}_{i,j}$ and $\chi^{(1)}_{i',j}$ that remain in the scope of any $\forall(\bar{\mathrm{x}}_n\cap X_j)$ exclusively contain literals from $\mathcal{L}_j$. Although the universal quantification in the last conjunct is vacuous, we write it here for the sake of clarity. We regroup the disjuncts in φ'' as follows

$$\varphi'' = \forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\ldots\exists\bar{\mathrm{y}}_{n-1}.\bigwedge_{S\in\mathcal{P}I\setminus\emptyset}\Big(\underbrace{\Big(\bigvee_{j=0}^{n-1}\forall(\bar{\mathrm{x}}_n\cap X_j).\bigvee_{i\in S}\chi^{(1)}_{i,j}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_j,\bar{\mathrm{x}}_{j+1},\ldots,\bar{\mathrm{x}}_n)\Big)}_{=:\,\chi^{(2)}_{S,j}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_j,\bar{\mathrm{x}}_{j+1},\ldots,\bar{\mathrm{x}}_{n-1})} \vee\underbrace{\Big(\forall(\bar{\mathrm{x}}_n\cap X_{n-1}).\textstyle\bigvee_{i\in S}\chi^{(1)}_{i,n-1}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_{n-1},\bar{\mathrm{x}}_n)\Big) \vee\Big(\bigvee_{f\in\mathcal{F}}\exists\bar{\mathrm{y}}_n.\bigwedge_{i\in S}L_{f(i)}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_n)\Big)\Big)}_{=:\,\chi^{(2)}_{S,\geq n-1}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_{n-1})}\,.$$

We now iterate these two steps in an alternating fashion until all quantifier blocks have been shifted inwards in the described way. The result $\varphi^{(2n)}$ has a tree-like shape with respect to the nesting of scopes of universal and existential quantifier blocks interspersed with conjunctions and disjunctions such that every atom $A(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_i,\bar{\mathrm{x}}_{i+1},\ldots,\bar{\mathrm{x}}_n)$ that belongs to the partition At_i lies exactly in the scope of the quantifier blocks $\exists\bar{\mathrm{y}}_1,\ldots,\exists\bar{\mathrm{y}}_i,\forall(\bar{\mathrm{x}}_{i+1}\cap X_i),\ldots,\forall(\bar{\mathrm{x}}_n\cap X_i)$. Figure 3.3 illustrates the situation in a simplified way.

We observe that every subformula $\forall x.\,\eta$ of $\varphi^{(2n)}$ that is not in the scope of any other universal quantifier has the shape

$$\forall(\bar{\mathrm{x}}_i\cap X_j).\bigvee_{\ell_1}\forall(\bar{\mathrm{x}}_{i+1}\cap X_j).\bigvee_{\ell_2}\Big(\ldots\Big(\bigvee_{\ell_{n-1}}\forall(\bar{\mathrm{x}}_n\cap X_j).\bigvee_{k\in K_{\ell_1,\ldots,\ell_{n-1}}}\eta'_k(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_j,\bar{\mathrm{x}}_{j+1},\ldots,\bar{\mathrm{x}}_n)\Big)\ldots\Big)\,,$$

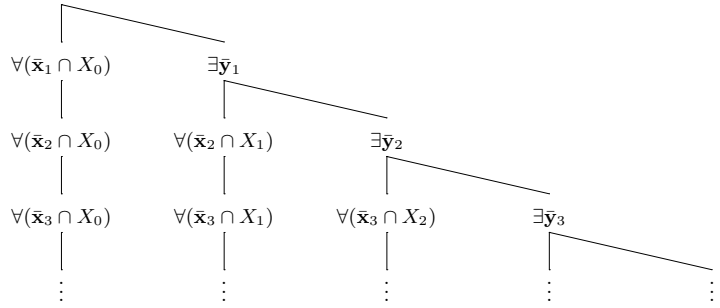


Figure 3.3: Nesting of quantifier blocks in the formula $\varphi^{(2n)}$.

for some j where the sets $K_{\ell_1,\ldots,\ell_{n-1}}$ are certain index sets. By a similar transformation as we have applied in the proof of Lemma 3.2.5 to transform the subformulas $\eta_{qp}^{(3)}$ into $\eta_{qp}^{(5)}$, we can exploit the pairwise separateness of the sets $V_1, \ldots, V_m$ in $\varphi^{(2n)}$ and transform the above subformula into

$$\bigvee_{1\leq h\leq m} \forall(\bar{\mathrm{x}}_i \cap X_j \cap V_h).\bigvee_{\ell'_1} \forall(\bar{\mathrm{x}}_{i+1} \cap X_j \cap V_h).\bigvee_{\ell'_2}\Big(\ldots \Big(\bigvee_{\ell_{n-1}} \forall(\bar{\mathrm{x}}_n \cap X_j \cap V_h).\bigvee_{k\in K^{(h)}_{\ell'_1,\ldots,\ell'_{n-1}}} \eta'_k(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_j,\bar{\mathrm{x}}_{j+1},\ldots,\bar{\mathrm{x}}_n)\Big)\ldots\Big),$$

where the $K^{(1)}_{\ell'_1,\ldots,\ell'_{n-1}},\ldots,K^{(m)}_{\ell'_1,\ldots,\ell'_{n-1}}$ are certain pairwise disjoint index sets. The resulting structure of quantifier nestings is depicted in Figure 3.4. Afterwards, we do a similar transformation for

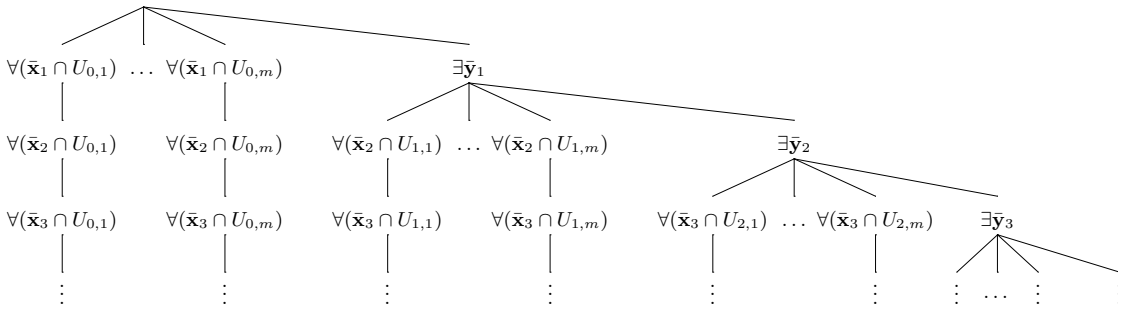


Figure 3.4: Illustration of the nesting of quantifier blocks in $\varphi^{(2n)}$ after narrowing the scopes of universal quantifiers with respect to the sets $V_1, \ldots, V_m$ that are pairwise separated in $\varphi^{(2n)}$. The sets $U_{j,h}$ denote the intersection $X_j \cap V_h$.

the existential quantifier blocks in $\varphi^{(2n)}$. Figure 3.5 depicts the resulting nesting structure of quantifiers. We denote the sentence resulting from $\varphi^{(2n)}$ after the described transformations by φ_*.

From this point on we argue along the same lines as in the proof of Lemma 3.2.5 to obtain an upper bound on the subformulas in φ_* that do not occur in the scope of any quantifiers. Let $\mathcal{L}_\varphi(V_h)$ denote the number of literals occurring in φ that contain at least one variable from V_h. Moreover, let κ be the least upper bound for all $|\mathcal{L}_\varphi(V_h)|$. Since the nesting depth of quantifiers in φ_* is bounded from above by $\partial_{\exists\forall}(\varphi)$, adapting the arguments from the proof of Lemma 3.2.5 entails that there are no more than $2^{\uparrow\partial_{\exists\forall}(\varphi)}(\kappa+n)$ different subformulas (not occurrences thereof!) of the mentioned kind. The term $\kappa+n$ (in contrast to only κ in the original proof) accounts for the fact that the subformulas may contain a certain mixture of existential and universal quantifiers.

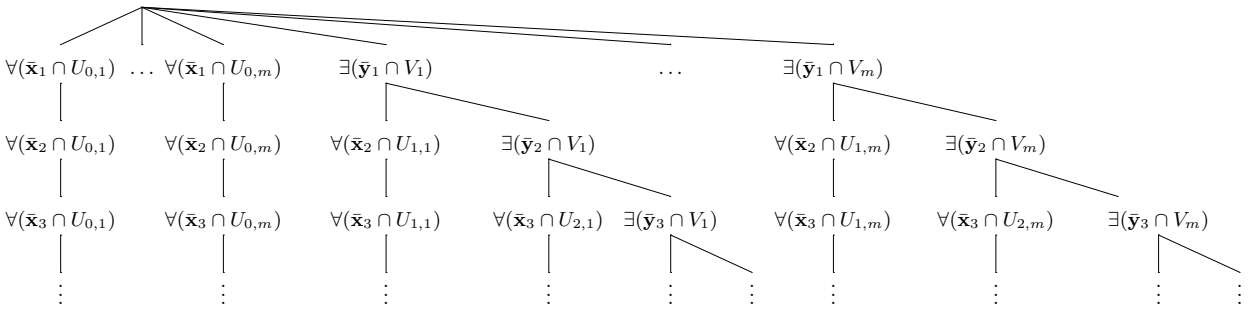


Figure 3.5: Illustration of the nesting of quantifier blocks in $\varphi^{(2n)}$ after narrowing the scopes of universal and existential quantifiers with respect to the pairwise-separated sets $V_1, \ldots, V_m$. The sets $U_{j,h}$ denote the intersection $X_j \cap V_h$.

More precisely, we now have $2^{2 \cdot 2^{2 \cdot 2^{\cdot^{\cdot^{\cdot^{2 \cdot 2^\kappa}}}}}}$ instead of $2^{2^{2^{\cdot^{\cdot^{\cdot^{2^\kappa}}}}}}$ different subformulas, and the former expression is bounded from above by $2^{\uparrow \partial_{\exists\forall}(\varphi)}(\kappa + n)$.

After shifting all quantifiers in φ_* outwards in an existential quantifiers-first manner, we obtain a BSR sentence that is equivalent to the original GBSR sentence φ and contains at most $|\bar{\mathbf{y}}|^2 \cdot \partial_{\exists\forall}(\varphi) \cdot \big(2^{\uparrow \partial_\exists(\varphi)}(\kappa + n)\big)^{\partial_{\exists\forall}(\varphi)}$ leading existential quantifiers. □

The just proven lemma can be reformulated into the following less technical theorem.

Theorem 3.5.3. *Every GBSR sentence is equivalent to some BSR sentence whose length is $\partial_{\exists\forall}(\varphi)$-fold exponential in the length of the original.*

The theorem also holds in the presence of constant symbols: every GBSR sentence φ with constant symbols is equivalent to some BSR sentence φ' with the same constant symbols.

After transforming a satisfiable GBSR sentence into an equivalent BSR sentence, the number of leading existential quantifiers induces an upper bound on the size of *small models* — every satisfiable GBSR sentence has such a small model. By virtue of Theorem 3.5.3, the small model property of BSR (with or without constant symbols), spelled out in Proposition 3.1.6, can be transferred to GBSR.

Corollary 3.5.4. *Every satisfiable GBSR sentence φ has a model whose size is at most $\partial_{\exists\forall}(\varphi)$-fold exponential in the length of φ. Moreover,* GBSR-Sat *is decidable, even if we allow constant symbols to occur.*

In Section 4.2 we present a different, a model-theoretic approach to GBSR-Sat which culminates in a direct construction of models. That approach facilitates deriving an upper bound on the size of small models as well. In order to formulate this bound accurately, we introduce a related, yet somewhat complementary notion of *degree* based on the interaction of universally quantified variables in atoms.

Regarding lower bounds, the result formulated in Theorem 3.2.7 immediately entails that there are GBSR sentences that inevitably lead to a non-elementary blowup when translating them into equivalent BSR sentences. Moreover, Theorem 3.3.18 is also relevant for GBSR. It means that for every natural number k there are GBSR sentences whose shortest equivalent in Gaifman normal form is k-fold exponentially longer than the original.

3.6 Taking Boolean Structure into Account

In this section we briefly look into the quest for gaining additional information from the Boolean structure of formulas. A trivial first observation in this context follows from Lemma 3.5.2 and the

fact that every $\wedge$-$\vee$-combination of BSR sentences is equivalent to some BSR sentence of the same length.

Proposition 3.6.1. *Every $\wedge$-$\vee$-combination of GBSR sentences is equivalent to some BSR sentence.*

Next, we increase the level of difficulty slowly and consider two special cases of GBSR sentences in conjunctive normal form (CNF) and in disjunctive normal form (DNF) where the translation into BSR does not lead to any blowup regarding the length of formulas.

Proposition 3.6.2. *Consider any GBSR sentence $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \bigwedge_{i=1}^{m} \psi_i$ where the ψ_i are disjunctions of literals. Let $\bar{\mathrm{y}} := \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_n$. Suppose that for any two distinct indices j, j' the sets $\mathrm{vars}(\psi_j) \cap \bar{\mathrm{y}}$ and $\mathrm{vars}(\psi_{j'}) \cap \bar{\mathrm{y}}$ are disjoint. Then, φ is equivalent to $\exists \bar{\mathrm{y}}_1 \ldots \bar{\mathrm{y}}_n \forall \bar{\mathrm{x}}_1 \ldots \bar{\mathrm{x}}_n . \bigwedge_{i=1}^{m} \psi_i$.*

Proof. By assumption, we can use quantifier shifting to transform φ into an equivalent sentence of the form $\bigwedge_{i=1}^{m} \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \psi_i$ — for simplicity, some instances of vacuous quantifiers have been introduced, i.e. quantifiers $\mathcal{Q} v . \chi$ where v does not occur in χ. The lemma follows immediately from the following auxiliary result.

Consider any GBSR sentence $\varphi' := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \psi$ in which ψ is a disjunction of literals. This means, we can rewrite ψ into a formula of the form

$$\chi_0(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n) \vee \chi_1(\bar{\mathrm{y}}_1, \bar{\mathrm{x}}_2, \ldots, \bar{\mathrm{x}}_n) \vee \ldots \vee \chi_{n-1}(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_{n-1}, \bar{\mathrm{x}}_n) \vee \chi_n(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n) \ ,$$

where none of the disjunctions χ_i, χ_j with $i \neq j$ share variables from $\bar{\mathrm{x}} := \bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_n$. For every
X_i i, $0 \leq i \leq n$, let $X_i := \mathrm{vars}(\chi_i) \cap \bar{\mathrm{x}}$. Then, φ' is equivalent to

$$\begin{aligned} &\big(\forall X_0 . \chi_0(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n)\big) \vee \big(\exists \bar{\mathrm{y}}_1 \forall X_1 . \chi_1(\bar{\mathrm{y}}_1, \bar{\mathrm{x}}_2, \ldots, \bar{\mathrm{x}}_n)\big) \\ &\qquad \vdots \\ &\vee \big(\exists \bar{\mathrm{y}}_1 \ldots \bar{\mathrm{y}}_{n-1} \forall X_{n-1} . \chi_{n-1}(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_{n-1}), \bar{\mathrm{x}}_n\big) \\ &\vee \big(\exists \bar{\mathrm{y}}_1 \ldots \bar{\mathrm{y}}_n . \chi_n(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n)\big) \ , \end{aligned}$$

which is a disjunction of BSR sentences. Shifting the quantifiers outwards in the right order yields the equivalent BSR sentence $\exists \bar{\mathrm{y}}_1 \ldots \bar{\mathrm{y}}_n \forall \bar{\mathrm{x}}_1 \ldots \bar{\mathrm{x}}_n . \psi$. □

As quantifier shifting is sufficient to transform the special kind of sentences treated in Proposition 3.6.2 into BSR sentences, the translation does not lead to a blowup in formula length. Hence, any satisfiable sentence φ of this kind has a model whose domain contains at most $\mathrm{len}(\varphi)$ elements.

Next, we briefly discuss the dual case of GBSR formulas in DNF where disjuncts do not share universally quantified variables.

Proposition 3.6.3. *Consider any GBSR sentence of the form $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \bigvee_{i=1}^{m} \psi_i$ where the ψ_i are conjunctions of literals. Let $\bar{\mathrm{x}} := \bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_n$ and suppose that for any two distinct indices j, j' the sets $\mathrm{vars}(\psi_j) \cap \bar{\mathrm{x}}$ and $\mathrm{vars}(\psi_{j'}) \cap \bar{\mathrm{x}}$ are disjoint. Then, φ is equivalent to $\exists \bar{\mathrm{y}}_1 \ldots \bar{\mathrm{y}}_n \forall \bar{\mathrm{x}}_1 \ldots \bar{\mathrm{x}}_n . \bigvee_{i=1}^{m} \psi_i$.*

Proof. Again, quantifier shifting and the introduction of vacuous quantifiers can be used to transform φ into an equivalent sentence of the form $\bigvee_{i=1}^{m} \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \psi_i$. The lemma follows immediately from the following auxiliary result.

Consider any GBSR sentence $\varphi = \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \psi$ in which ψ is a conjunction of literals. Again, we can regroup the literals in this conjunction so that ψ has the form

$$\chi_0(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n) \wedge \chi_1(\bar{\mathrm{y}}_1, \bar{\mathrm{x}}_2, \ldots, \bar{\mathrm{x}}_n) \wedge \ldots \wedge \chi_{n-1}(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_{n-1}, \bar{\mathrm{x}}_n) \wedge \chi_n(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n) \ .$$

Then, φ is equivalent to

$$\begin{aligned}\big(\forall \bar{\mathrm{x}}_1 \ldots \bar{\mathrm{x}}_n.\, \chi_0(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n)\big) \wedge \Big(\exists \bar{\mathrm{y}}_1.\, \big(\forall \bar{\mathrm{x}}_2 \ldots \bar{\mathrm{x}}_n.\, \chi_1(\bar{\mathrm{y}}_1, \bar{\mathrm{x}}_2, \ldots, \bar{\mathrm{x}}_n)\big) \\ \wedge \Big(\exists \bar{\mathrm{y}}_2.\, \ldots \\ \wedge \Big(\exists \bar{\mathrm{y}}_{n-1}.\, \big(\forall \bar{\mathrm{x}}_n.\, \chi_{n-1}(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_{n-1}, \bar{\mathrm{x}}_n)\big) \\ \wedge \Big(\exists \bar{\mathrm{y}}_n.\, \chi_n(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n)\Big)\Big) \ldots\Big)\Big)\,.\end{aligned}$$

From this we obtain the BSR sentence $\exists \bar{\mathrm{y}}_1 \ldots \bar{\mathrm{y}}_n \forall \bar{\mathrm{x}}_1 \ldots \bar{\mathrm{x}}_n.\, \psi$ by shifting quantifiers outwards in the right order. □

Both propositions refer to special cases of GBSR sentences where the translation into BSR requires very little effort, as only quantifiers need to be shifted. Moreover, the sentences are required to possess a very specific Boolean structure. The latter requirement can be weakened to some extent. As any quantifier-free formula can be converted into conjunctive or disjunctive normal form, one could simply extend Propositions 3.6.2 and 3.6.3 to formulas that can be transformed into formulas of the described shape. In other words, the original formula need not satisfy the requirements, but a certain normal form has to. But then, checking whether a given formula falls into the syntactic category in question may require exponential time, in the worst case, as the normal form transformation can cause an exponential blowup regarding formula length.

Alternatively, we can use an approximation of the normal form that yields enough information to make an informed decision without requiring the expensive construction of conjunctive or disjunctive normal forms.[4] Suppose we are given a quantifier-free formula ψ in negation normal form. From its Boolean structure we can read off which literals will end up in a common conjunction when we transform ψ into DNF using exclusively the basic laws of Boolean algebra: associativity, commutativity, distributivity of $\wedge$ over $\vee$ — the latter is only used in the direction from $\chi_1 \wedge (\chi_2 \vee \chi_3)$ to $(\chi_1 \wedge \chi_2) \vee (\chi_1 \wedge \chi_3)$. Each application of these rules preserves the following property: we call two atoms A, B *conjunctive companions in* ψ, if ψ contains a subformula $\chi_1 \wedge \chi_2$ such that A occurs in χ_1 and B occurs in χ_2 or vice versa. There is also the dual notion of *disjunctive companions in* ψ, which applies to atoms A, B if ψ contains a subformula $\chi_1 \vee \chi_2$ such that A occurs in χ_1 and B occurs in χ_2 or vice versa.

conjunctive and disjunctive companions

Lemma 3.6.4 (Invariance of conjunctive and disjunctive companions)**.**

(i) Let $\psi := (\chi_1 \circ \chi_2) \circ \chi_3$ and $\psi' := \chi_1 \circ (\chi_2 \circ \chi_3)$ be formulas with $\circ \in \{\wedge, \vee\}$. Two atoms A and B are conjunctive (disjunctive) companions in ψ if and only if they are conjunctive (disjunctive) companions in ψ'.

(ii) Let $\psi := \chi_1 \circ \chi_2$ and $\psi' := \chi_2 \circ \chi_1$ be formulas with $\circ \in \{\wedge, \vee\}$. Two atoms A and B are conjunctive (disjunctive) companions in ψ if and only if they are conjunctive (disjunctive) companions in ψ'.

(iii) Let $\psi := \chi_1 \wedge (\chi_2 \vee \chi_3)$ and $\psi' := (\chi_1 \wedge \chi_2) \vee (\chi_1 \wedge \chi_3)$. Two atoms A and B are conjunctive companions in ψ if and only if they are conjunctive companions in ψ'.

(iv) Let $\psi := \chi_1 \vee (\chi_2 \wedge \chi_3)$ and $\psi' := (\chi_1 \vee \chi_2) \wedge (\chi_1 \vee \chi_3)$. Two atoms A and B are disjunctive companions in ψ if and only if they are disjunctive companions in ψ'.

Proof Sketch. The proof is straightforward and proceeds by case distinction with respect to the subformulas in which A and B occur. □

[4] The approximation scheme we discuss here was already used in [Kos16], Section 3.2, under the label *conjunctive associativity*. Košta makes use of the concept in order to substantially reduce the size of elimination sets in the context of quantifier elimination by virtual substitution. Basic ideas in this direction were already present in [Dol00].

For every quantifier-free formula ψ in negation normal form Lemma 3.6.4 entails that two atoms are conjunctive companions in ψ if and only if they are conjunctive companions in an equivalent formula ψ' in DNF, provided that ψ' has been derived from ψ by applying exclusively the Boolean laws of associativity, commutativity, and distributivity of $\wedge$ over $\vee$. Dually, two atoms are disjunctive companions in ψ if and only if they are disjunctive companions in an equivalent formula ψ'' in CNF, provided that ψ'' has been derived from ψ using exclusively the Boolean laws of associativity, commutativity, and distributivity of $\vee$ over $\wedge$. This leads to the following observation.

Definition 3.6.5 (Conjunctively and Disjunctively Connected Sets of Variables and Atoms)**.** *Consider any two sets X, Y of first-order variables and any formula ψ in negation normal form. We say that X and Y are* conjunctively (disjunctively) connected in ψ*, if there is some $x \in X \cap Y$ that occurs in ψ, or if there are two atoms A and B that are conjunctive (disjunctive) companions in ψ such that A contains some $x \in X$ and B contains some $y \in Y$. We say that two sets X, Y of variables are* conjunctively (disjunctively) disconnected in ψ*, if they are not conjunctively (disjunctively) connected in ψ.*

We extend these notions to mixed pairs X, S of sets of first-order variables and of atoms, respectively: X and S are conjunctively (disjunctively) connected in ψ*, if there exists some $x \in X$ and some $A \in S$ that occurs in ψ and contains x, or if there are two atoms A and B that are conjunctive (disjunctive) companions in ψ such that A contains some $x \in X$ and B belongs to S. We say that X, S are* conjunctively (disjunctively) disconnected in ψ*, if they are not conjunctively (disjunctively) connected in ψ.*

Lemma 3.6.6. *Consider any first-order sentence of the form $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \psi$ in negation normal form with quantifier-free ψ. If for every i, $1 \leq i \leq n$, the sets $\bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_i$ and $\bar{\mathrm{y}}_i$ are disjunctively disconnected in ψ, then φ is equivalent to $\exists \bar{\mathrm{y}}_1 \ldots \bar{\mathrm{y}}_n \forall \bar{\mathrm{x}}_1 \ldots \bar{\mathrm{x}}_n . \psi$.*

Proof. By Lemma 3.6.4, we can transform ψ into a formula ψ' in CNF that is a conjunction of clauses none of which contains variables from two disjunctively disconnected sets. Hence, ψ' can be rewritten into the form

$$\Big(\bigwedge_{i \in I_0} \chi_i(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n)\Big) \wedge \Big(\bigwedge_{i \in I_1} \chi_i(\bar{\mathrm{y}}_1, \bar{\mathrm{x}}_2, \ldots, \bar{\mathrm{x}}_n)\Big) \wedge \ldots \wedge \Big(\bigwedge_{i \in I_n} \chi_i(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n)\Big) \ ,$$

where the index sets $I_0, \ldots, I_n$ are finite and pairwise disjoint, and every χ_i is a clause. Using quantifier shifting, we can transform $\forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \psi'$ into the sentence

$$\begin{aligned}
&\big(\forall \bar{\mathrm{x}}_1 \ldots \bar{\mathrm{x}}_n . \bigwedge_{i \in I_0} \chi_i(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n)\big) \\
&\wedge \Big(\exists \bar{\mathrm{y}}_1 . \big(\forall \bar{\mathrm{x}}_2 \ldots \bar{\mathrm{x}}_n . \bigwedge_{i \in I_1} \chi_i(\bar{\mathrm{y}}_1, \bar{\mathrm{x}}_2, \ldots, \bar{\mathrm{x}}_n)\big) \\
&\qquad \wedge \Big(\exists \bar{\mathrm{y}}_2 . \ \ldots \\
&\qquad\qquad \wedge \Big(\exists \bar{\mathrm{y}}_{n-1} . \big(\forall \bar{\mathrm{x}}_n . \bigwedge_{i \in I_{n-1}} \chi_i(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_{n-1}, \bar{\mathrm{x}}_n)\big) \\
&\qquad\qquad\qquad \wedge \Big(\exists \bar{\mathrm{y}}_n . \bigwedge_{i \in I_n} \chi_i(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n)\Big)\Big) \ldots\Big)\Big) \ .
\end{aligned}$$

If we now shift quantifiers outwards in the right order, we obtain $\exists \bar{\mathrm{y}}_1 \ldots \bar{\mathrm{y}}_n \forall \bar{\mathrm{x}}_1 \ldots \bar{\mathrm{x}}_n . \psi'$, which is equivalent to $\exists \bar{\mathrm{y}}_1 \ldots \bar{\mathrm{y}}_n \forall \bar{\mathrm{x}}_1 \ldots \bar{\mathrm{x}}_n . \psi$. □

The idea underlying Lemma 3.6.6 and the definition of GBSR can be blended to obtain a more liberal definition of GBSR. In particular, this helps liberalizing requirement (ii) of Definition 3.4.1 up to a certain degree. The following lemma illustrates a first approach to such a blend of separateness and disjunctive disconnectedness. Several refinements would be conceivable, but we shall confine ourselves to this rather simple variant.

Lemma 3.6.7. *Let $\varphi := \forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1 \ldots \forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n.\, \psi$ be any relational first-order sentence with equality. Let* At *be the set of all atoms occurring in φ and let $\bar{\mathrm{x}} := \bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_n$ and $\bar{\mathrm{y}} := \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_n$. Assume that there is a sequence of subsets $\mathrm{At}^{(0)}, \mathrm{At}^{(1)}, \ldots, \mathrm{At}^{(n)} \subseteq \mathrm{At}$ such that for every i, $1 \leq i \leq n$, the set $\bar{\mathrm{y}}_i$ is disjunctively disconnected from the set $\mathrm{At}^{(0)} \cup \ldots \cup \mathrm{At}^{(i-1)}$ in ψ. Moreover, assume that for every k, $0 \leq k \leq n$, the set $\mathrm{At}^{(k)}$ can be partitioned into parts $\mathrm{At}_0^{(k)}, \ldots, \mathrm{At}_k^{(k)}$ such that* $\mathrm{At}^{(k)}$

(i) for every i, $0 \leq i \leq k$, we have $\mathrm{vars}(\mathrm{At}_i^{(k)}) \subseteq \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_i \cup \bar{\mathrm{x}}_{i+1} \cup \ldots \cup \bar{\mathrm{x}}_n$, and

(ii) for all distinct i, j, $0 \leq i < j \leq k$, we have $\mathrm{vars}(\mathrm{At}_i^{(k)}) \cap \mathrm{vars}(\mathrm{At}_j^{(k)}) \cap \bar{\mathrm{x}} = \emptyset$, and

(iii) for all distinct k, ℓ and all distinct i, j with $0 \leq k < \ell \leq n$ and $0 \leq i \leq k$ and $0 \leq j \leq \ell$ we have
$$\mathrm{vars}(\mathrm{At}_i^{(k)}) \cap \mathrm{vars}(\mathrm{At}_j^{(\ell)}) \cap \big(\bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_k\big) = \emptyset.$$

Then, φ is equivalent to some GBSR sentence.

Proof. By Lemma 3.6.4, we can transform ψ into a formula ψ' in CNF that can be rewritten into the form

$$\psi'' := \ \chi_0(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n) \wedge \chi_1(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1) \wedge \ldots \wedge \chi_n(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n)\ ,$$

where the χ_k are (possibly empty) conjunctions of clauses, exclusively containing atoms from $\mathrm{At}^{(k)}$. Straightforward quantifier shifting allows us to transform $\forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1 \ldots \forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n.\, \psi''$ into the sentence

$$\begin{aligned}
&\big(\forall\bar{\mathrm{x}}_1 \ldots \bar{\mathrm{x}}_n.\, \chi_0(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n)\big) \\
&\wedge \Big(\forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1.\, \big(\forall\bar{\mathrm{x}}_2 \ldots \bar{\mathrm{x}}_n.\, \chi_1(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1)\big) \\
&\qquad \wedge \Big(\forall\bar{\mathrm{x}}_2\exists\bar{\mathrm{y}}_2.\ \ldots \\
&\qquad\quad \wedge \Big(\forall\bar{\mathrm{x}}_{n-1}\exists\bar{\mathrm{y}}_{n-1}.\, \big(\forall\bar{\mathrm{x}}_n.\, \chi_{n-1}(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_{n-1})\big) \\
&\qquad\qquad \wedge \Big(\forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n.\, \chi_n(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n)\Big)\Big) \ldots\Big)\Big)\ .
\end{aligned}$$

Renaming some of the bound variables yields

$$\begin{aligned}
&\varphi'' := \\
&\big(\forall\bar{\mathrm{x}}_1^{(0)} \ldots \bar{\mathrm{x}}_n^{(0)}.\, \chi_0(\bar{\mathrm{x}}_1^{(0)}, \ldots, \bar{\mathrm{x}}_n^{(0)})\big) \\
&\wedge \Big(\forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1.\, \big(\forall\bar{\mathrm{x}}_2^{(1)} \ldots \bar{\mathrm{x}}_n^{(1)}.\, \chi_1(\bar{\mathrm{x}}_1, \bar{\mathrm{x}}_2^{(1)}, \ldots, \bar{\mathrm{x}}_n^{(1)}, \bar{\mathrm{y}}_1)\big) \\
&\qquad \wedge \Big(\forall\bar{\mathrm{x}}_2\exists\bar{\mathrm{y}}_2.\, \big(\forall\bar{\mathrm{x}}_3^{(2)} \ldots \bar{\mathrm{x}}_n^{(2)}.\, \chi_2(\bar{\mathrm{x}}_1, \bar{\mathrm{x}}_2, \bar{\mathrm{x}}_3^{(2)}, \ldots, \bar{\mathrm{x}}_n^{(2)}, \bar{\mathrm{y}}_1, \bar{\mathrm{y}}_2)\big) \\
&\qquad\quad \wedge \Big(\forall\bar{\mathrm{x}}_3\exists\bar{\mathrm{y}}_3.\ \ldots \\
&\qquad\qquad \wedge \Big(\forall\bar{\mathrm{x}}_{n-1}\exists\bar{\mathrm{y}}_{n-1}.\, \big(\forall\bar{\mathrm{x}}_n^{(n-1)}.\, \chi_{n-1}(\bar{\mathrm{x}}_1, \bar{\mathrm{x}}_2, \bar{\mathrm{x}}_3, \ldots, \bar{\mathrm{x}}_{n-1}, \bar{\mathrm{x}}_n^{(n-1)}, \bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_{n-1})\big) \\
&\qquad\qquad\quad \wedge \Big(\forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n.\, \chi_n(\bar{\mathrm{x}}_1, \bar{\mathrm{x}}_2, \ldots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n)\Big)\Big) \ldots\Big)\Big)\ .
\end{aligned}$$

For every k, $0 \leq k \leq n$, we define $\widetilde{\mathrm{At}}^{(k)}$ to be the set of atoms occurring in the formula $\widetilde{\mathrm{At}}^{(k)}$

$$\chi_k(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_k, \bar{\mathrm{x}}_{k+1}^{(k)}, \ldots, \bar{\mathrm{x}}_k^{(k)}, \bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_k)\ .$$

In other words, $\widetilde{\mathrm{At}}^{(k)}$ is a subset of the atoms from $\mathrm{At}^{(k)}$ after renaming variables like in φ'''. Due to our assumptions, every $\widetilde{\mathrm{At}}^{(k)}$ can be partitioned into sets $\widetilde{\mathrm{At}}_0^{(k)}, \ldots, \widetilde{\mathrm{At}}_k^{(k)}$ such that $\widetilde{\mathrm{At}}_i^{(k)}$

(a) for every i, $0 \leq i \leq k$, we have $\mathrm{vars}\big(\widetilde{\mathrm{At}}_i^{(k)}\big) \subseteq \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_i \cup \bar{\mathrm{x}}_{i+1} \cup \ldots \cup \bar{\mathrm{x}}_k \cup \bar{\mathrm{x}}_{k+1}^{(k)} \cup \ldots \cup \bar{\mathrm{x}}_n^{(k)}$,

(b) for all distinct i, j, $0 \leq i < j \leq k$, we have $\mathrm{vars}\big(\widetilde{\mathrm{At}}_i^{(k)}\big) \cap \mathrm{vars}\big(\widetilde{\mathrm{At}}_j^{(k)}\big) \cap \big(\bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_k \cup \bar{\mathrm{x}}_{k+1}^{(k)} \cup \ldots \cup \bar{\mathrm{x}}_n^{(k)}\big) = \emptyset$, and

(c) for all distinct k, ℓ and all distinct i, j with $0 \leq k < \ell \leq n$ and $0 \leq i \leq k$ and $0 \leq j \leq \ell$ we have $\mathrm{vars}(\widetilde{\mathrm{At}}_i^{(k)}) \cap \mathrm{vars}(\widetilde{\mathrm{At}}_j^{(\ell)}) \cap \Big(\bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_\ell \cup \bigcup_{k \leq k' \leq n} \big(\bar{\mathrm{x}}_{k'+1}^{(k')} \cup \ldots \cup \bar{\mathrm{x}}_n^{(k')}\big)\Big) = \emptyset$.

$\widetilde{\mathrm{At}}_i$

Consider the sets $\widetilde{\mathrm{At}}_i := \bigcup_{i \leq k \leq n} \widetilde{\mathrm{At}}_i^{(k)}$ for $0 \leq i \leq n$. Because of (a), we have

$$\mathrm{vars}\big(\widetilde{\mathrm{At}}_i\big) \subseteq \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_i \cup \bar{\mathrm{x}}_{i+1} \cup \ldots \cup \bar{\mathrm{x}}_n \cup \bigcup_{i \leq k \leq n} \big(\bar{\mathrm{x}}_{k+1}^{(k)} \cup \ldots \cup \bar{\mathrm{x}}_n^{(k)}\big) \ . \tag{3.2}$$

Together with (b) and (c), this entails

$$\mathrm{vars}(\widetilde{\mathrm{At}}_i) \cap \mathrm{vars}(\widetilde{\mathrm{At}}_j) \cap \Big(\bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_n \cup \bigcup_{1 \leq k \leq n} \big(\bar{\mathrm{x}}_{k+1}^{(k)} \cup \ldots \cup \bar{\mathrm{x}}_n^{(k)}\big)\Big) = \emptyset \tag{3.3}$$

for all i, j with $0 \leq i < j \leq n$.

Let φ''' be the following formula that results from φ'' by shifting all quantifiers outward:

$$\begin{aligned}\varphi''' := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n \forall \bar{\mathrm{x}}_1^{(0)} \ldots \bar{\mathrm{x}}_n^{(0)} \bar{\mathrm{x}}_2^{(1)} \ldots \bar{\mathrm{x}}_n^{(1)} \ldots \bar{\mathrm{x}}_{n-1}^{(n-2)} \bar{\mathrm{x}}_n^{(n-2)} \bar{\mathrm{x}}_n^{(n-1)}. \\ \chi_0(\bar{\mathrm{x}}_1^{(0)}, \ldots, \bar{\mathrm{x}}_n^{(0)}) \wedge \chi_1(\bar{\mathrm{x}}_1, \bar{\mathrm{x}}_2^{(1)}, \ldots, \bar{\mathrm{x}}_n^{(1)}, \bar{\mathrm{y}}_1) \wedge \ldots \\ \wedge \chi_{n-1}(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_{n-1}, \bar{\mathrm{x}}_n^{(n-1)}, \bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_{n-1}) \wedge \chi_n(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n) \ .\end{aligned}$$

By virtue of (3.2) and (3.3), φ''' is the sought GBSR sentence equivalent to φ □

Notice that the maximal possible degree of interaction in the sentence φ''' in the proof of Lemma 3.6.7 is $n+1$. However, we could do the shifting of quantifiers in the last step of φ''''s construction in such a way that $\partial_{\exists\forall}(\varphi''') \leq n$. Consequently, the blowup that we incur when transforming sentences that satisfy the conditions of Lemma 3.6.7 is at most $(n+1)$-fold exponential in the length of φ, since φ''''s matrix might be exponentially longer than the original matrix of φ.

Example 3.6.8. *Consider the sentence*

$$\varphi := \forall x_1 x_2 \exists y \forall z_1 z_2. \big(\big(P(x_1, z_1) \vee P(z_2, x_2)\big) \wedge P(y, z_1)\big) \vee \big(P(x_2, z_2) \wedge P(x_1, x_2)\big) \ ,$$

which satisfies the conditions of Lemma 3.6.7. To see this, we inspect the following witnessing subsets $\mathrm{At}^{(0)}$ *and* $\mathrm{At}^{(1)}$ *with their respective partitions:*

$$\mathrm{At}^{(0)} := \big\{\underbrace{P(x_1, z_1), P(z_2, x_2)}_{=: \mathrm{At}_0^{(0)}}\big\} \qquad \textit{and} \qquad \mathrm{At}^{(1)} := \big\{\underbrace{P(y, z_1)}_{=: \mathrm{At}_1^{(1)}}, \underbrace{P(x_2, z_2), P(x_1, x_2)}_{=: \mathrm{At}_0^{(1)}}\big\} \ .$$

Proceeding as described in the proof of Lemma 3.6.7, the sentence φ *can be transformed into the equivalent sentence*

$$\begin{aligned}&\Big(\forall x_1 x_2 z_1 z_2. \big(P(x_1, z_1) \vee P(z_2, x_2) \vee P(x_2, z_2)\big) \wedge \big(P(x_1, z_1) \vee P(z_2, x_2) \vee P(x_1, x_2)\big)\Big) \\ &\wedge \Big(\forall x_1 x_2 \exists y \forall z_1 z_2. \big(P(y, z_1) \vee P(x_2, z_2)\big) \wedge \big(P(y, z_1) \vee P(x_1, x_2)\big)\Big) \ .\end{aligned}$$

Evidently, each of the two constituents of the topmost conjunction is a GBSR sentence.

3.7 The Generalized Ackermann Fragment (GAF)

Recall that the Ackermann fragment (AF) consists of all relational first-order sentences in prenex normal form with an $\exists^*\forall\exists^*$ quantifier prefix and without equality. In the beginning of Chapter 3, we have already outlined that the satisfiability problem for AF is decidable and that this decidability result has been stretched to several syntactic extensions of AF: AF with equality, the Gurevich–Maslov–Orevkov fragment — AF plus function symbols of arbitrary arity, and the Shelah fragment — AF with equality plus a single unary function symbol. Of course, constant symbols may also be allowed without jeopardizing decidability of the respective decidability problem. The Shelah fragment has the remarkable property of having a decidable satisfiability problem while allowing the formulation of infinity axioms. Hence, this class of sentences does not enjoy the finite model property.

In the present section, we generalize AF to the *generalized Ackermann fragment (GAF)* in the same spirit as we have generalizeed BSR to GBSR in Section 3.4. This means we shall devise an effective procedure that translates any given GAF sentence into an equivalent AF sentence. It will turn out that this procedure will be compatible with function symbols and equality. That is, our results will show that GAF with equality is equivalent to AF with equality, GAF with arbitrary function symbols but without equality is equivalent to the Gurevich–Maslov–Orevkov fragment, and GAF with equality and a single unary function symbol in equivalent to the Shelah fragment. Hence, all these extensions of GAF will be shown to possess a decidable satisfiability problem.

Intuitively speaking, a GAF sentence is of the form $\varphi := \forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\bar{\mathrm{u}}_1\ldots\forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n\bar{\mathrm{u}}_n.\,\psi$ with quantifier-free ψ and it satisfies the following properties. Each atom in φ contains only variables from some subsequence of φ's quantifier prefix of the form $\exists^*\forall\exists^*$. If two atoms share a universally quantified variable or some variable from the trailing $\exists^*$-block of their respective quantifier subsequence, then they have the same $\exists^*\forall\exists^*$-subsequence as source of all their variables.

Definition 3.7.1 (Generalized Ackermann fragment (GAF))**.** *Let* $\varphi := \forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\bar{\mathrm{u}}_1\ldots\forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n\bar{\mathrm{u}}_n.\,\psi$ *be a relational first-order sentence without equality. Let* At *be the set of all atoms occurring in* φ *and let* $\bar{\mathrm{x}} := \bar{\mathrm{x}}_1\cup\ldots\cup\bar{\mathrm{x}}_n$, $\bar{\mathrm{y}} := \bar{\mathrm{y}}_1\cup\ldots\cup\bar{\mathrm{y}}_n$, *and* $\bar{\mathrm{u}} := \bar{\mathrm{u}}_1\cup\ldots\cup\bar{\mathrm{u}}_n$. *Moreover, we define the* index of a variable $v\in\bar{\mathrm{x}}\cup\bar{\mathrm{y}}\cup\bar{\mathrm{u}}$ *by* $\mathrm{idx}(v) := k$ *if and only if* $v\in\bar{\mathrm{x}}_k\cup\bar{\mathrm{y}}_k\cup\bar{\mathrm{u}}_k$. *The sentence* φ *belongs to the* generalized Ackermann fragment (GAF) *if and only if we can partition* At *into sets* At_0 *and* At_x, $x\in\bar{\mathrm{x}}$, *such that the following conditions are satisfied:* $\mathrm{idx}(v)$
(a) $\mathrm{vars}(\mathrm{At}_0)\subseteq\bar{\mathrm{y}}$
(b) for every $x\in\bar{\mathrm{x}}$ *we have* $\mathrm{vars}(\mathrm{At}_x)\subseteq\bar{\mathrm{y}}_1\cup\ldots\cup\bar{\mathrm{y}}_{\mathrm{idx}(x)-1}\cup\{x\}\cup\bar{\mathrm{u}}_{\mathrm{idx}(x)}\cup\ldots\cup\bar{\mathrm{u}}_n$
(c) for all distinct $x,x'\in\bar{\mathrm{x}}$ *we have* $\mathrm{vars}(\mathrm{At}_x)\cap\mathrm{vars}(\mathrm{At}_{x'})\cap\bar{\mathrm{u}}=\emptyset$.

Notice that the tuples $\bar{\mathrm{x}}_i$ and $\bar{\mathrm{y}}_i,\bar{\mathrm{u}}_i$ in any GAF sentence φ may be empty. As one consequence, φ's quantifier prefix does not have to start with a universal quantifier and it does not have to end with an existential quantifier. Moreover, notice that every variable $u\in\bar{\mathrm{u}}$ that occurs in φ is associated with exactly one *reference variable* $x\in\bar{\mathrm{x}}$, determined by the set At_x in which u occurs. Intuitively speaking, using suitable equivalence-preserving transformations, any quantifier $\exists u$ with $u\in\bar{\mathrm{u}}$ can be shifted out of the scope of any universal quantifier but the one binding u's reference variable. This is the essence of the first step of the effective translation procedure from GAF into AF, which we shall assemble in the proof of Lemma 3.8.4. The following example gives a first impression of GAF sentences and how they can be translated into AF.

Example 3.7.2. *Consider the first-order sentence*

$$\begin{aligned}\varphi := \ \exists u\forall x\exists v\forall z\exists y_1y_2.\,\big(\neg P(u,x)\vee\big(Q(x,v)\wedge R(u,z,y_1)\big)\big)\\ \wedge\ \big(P(u,x)\vee\big(\neg Q(x,v)\wedge\neg R(u,z,y_2)\big)\big)\ .\end{aligned}$$

The partition of the set $\mathrm{At} := \{P(u,x),Q(x,v),R(u,z,y_1),R(u,z,y_2)\}$ *into* $\mathrm{At}_0 := \emptyset$, $\mathrm{At}_x := \{P(u,x),Q(x,v)\}$, *and* $\mathrm{At}_z := \{R(u,z,y_1),R(u,z,y_2)\}$ *is a witness for the belonging of* φ *to GAF. Due to the Boolean structure of* φ, *the quantifiers* $\exists y_2$, $\exists y_1$, *and* $\forall z$ *can be shifted inwards*

immediately but $\exists v$ cannot. This yields the equivalent sentence

$$\exists u \forall x \exists v. \big(\neg P(u,x) \vee \big(Q(x,v) \wedge \forall z \exists y_1. R(u,z,y_1)\big)\big) \\ \wedge \ \big(P(u,x) \vee \big(\neg Q(x,v) \wedge \forall z \exists y_2. \neg R(u,z,y_2)\big)\big) \ .$$

Because of the two universal quantifiers $\forall x$ and $\forall z$, which are even interspersed with an existential one, φ does not belong to AF. Exhaustive Skolemization of φ leads to

$$\forall x z. \big(\neg P(c,x) \vee \big(Q\big(x, f(x)\big) \wedge R\big(c,z,g(x,z)\big)\big)\big) \wedge \big(P(c,x) \vee \big(\neg Q\big(x, f(x)\big) \wedge \neg R\big(c,z,h(x,z)\big)\big)\big)$$

and thus explicitly fixes the dependence of y_1 on the universally quantified variables x and z, as y_1 is replaced with the term $g(x,z)$. However, the shape of the original φ did not immediately indicate such a strong dependence of y_1 on x, since x and y_1 do not co-occur in any atom. Moreover, there are no other variables that depend on x and establish a connection between x and y_1 by means of co-occurrences in atoms. One may say that it is the Boolean structure of φ alone which causes a dependence of y_1 on x, and that such a form of dependence has only a finite character*. These ideas will be made more precise in Chapter 4 and, for GAF sentences in particular, in Section 4.3.*

The described point of view is supported by the existence of an equivalent sentence φ', in which the dependence of y_1 on x has vanished. The price we have to pay, however, is an increase in the size of the formula.

$$\begin{aligned} \varphi' := \exists u. \Big(\forall x. \big(\neg P(u,x) \vee \exists v. Q(x,v)\big)\Big) \ \wedge \ \Big(\big(\forall x. \neg P(u,x)\big) \vee \forall z \exists y_1. R(u,z,y_1)\Big) \\ \wedge \ \Big(\forall x. \big(\exists v. \neg Q(x,v)\big) \vee P(u,x)\Big) \ \wedge \ \Big(\big(\forall x \exists v. \neg Q(x,v)\big) \vee \forall z \exists y_1. R(u,z,y_1)\Big) \\ \wedge \ \Big(\big(\forall z \exists y_2. \neg R(u,z,y_2)\big) \vee \forall x. P(u,x)\Big) \ \wedge \ \Big(\big(\forall z \exists y_2. \neg R(u,z,y_2)\big) \vee \forall x \exists v. Q(x,v)\Big) \\ \wedge \ \Big(\big(\forall z \exists y_2. \neg R(u,z,y_2)\big) \vee \forall z \exists y_1. R(u,z,y_1)\Big) \end{aligned}$$

Transforming φ into φ' requires only basic logical laws and is very similar to approaches we have seen before: first, we shift the quantifiers $\exists y_2, \exists y_1, \forall z$ inwards as far as possible. Then, we construct a disjunction of conjunctions of certain subformulas using distributivity. This allows us to shift the quantifier $\exists v$ inwards. Afterwards, we apply the laws of distributivity again to obtain a conjunction of disjunctions of certain subformulas. This step enables us to shift the universal quantifier $\forall x$ inwards. In the resulting sentence every occurrence of an existential quantifier lies in the scope of at most one universal quantifier. Moreover, every atom in the original formula φ contains at most one universally quantified variable. Exhaustive Skolemization of φ' leads to a sentence whose shape is quite close to the shape of an exhaustively Skolemized sentence from the Ackermann fragment. More precisely, every atom contains at most one variable, possibly with multiple occurrences. The only difference is that we get more than only one universally quantified variable in the sentence as a whole, but at most one in every atom.

Another example of a simple GAF sentence is the sentence $\psi := \exists u \forall x \exists y \forall z. \big(P(u,z) \wedge Q(u,x)\big) \vee \big(P(y,z) \wedge Q(u,y)\big)$ which we have already treated in Example 3.4.2 as an example for GBSR sentences.

The sentence ψ from the above example belongs to GBSR and GAF at the same time, while it does not belong to the Ackermann fragment, SF, BSR, or the monadic fragment. Hence, even the intersection of GBSR and GAF contains sentences which do not fall into the syntactic categories offered by the standard fragments.

As a first result concerning GAF we show that membership in GAF is decidable in polynomial time.

Theorem 3.7.3. *Deciding whether a given first-order sentence belongs to GAF can be done deterministically in time that is polynomial in the length of any reasonable encoding of the input sentence.*

Proof sketch. Let $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{v}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{v}}_n . \psi$ be any relational first-order sentence in prenex normal form with quantifier-free ψ. Let $\bar{\mathrm{x}} := \bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_n$ and $\bar{\mathrm{v}} := \bar{\mathrm{v}}_1 \cup \ldots \cup \bar{\mathrm{v}}_n$. For the moment we do not know a priori how the variables in each and every existential quantifier block $\exists \bar{\mathrm{v}}_k$ are to be partitioned into $\bar{\mathrm{y}}_k$ and $\bar{\mathrm{u}}_k$. This will be sorted out in due course.

Let $\mathcal{G}_\varphi := \langle V, E \rangle$ be a directed graph such that $V := \bar{\mathrm{v}}$ and $E := \big\{ \langle v, v' \rangle \;\big|\; \mathrm{idx}(v) \leq \mathrm{idx}(v') \text{ and there is some atom } A \text{ in } \psi \text{ in which } v \text{ and } v' \text{ co-occur} \big\}$. For any variable $v \in \bar{\mathrm{v}}$ the *upward closure* $\mathcal{C}^\uparrow_v$ is the smallest subset of $\bar{\mathrm{v}}$ such that $v \in \mathcal{C}^\uparrow_v$ and for every $v' \in \mathcal{C}^\uparrow_v$ the existence of an edge $\langle v', v'' \rangle$ in $\mathcal{G}_\varphi$ entails $v'' \in \mathcal{C}^\uparrow_v$. Let $\mathrm{at}(\mathcal{C}^\uparrow_v)$ denote the set of all atoms in ψ, in which a variable from $\mathcal{C}^\uparrow_v$ occurs. For every $x \in \bar{\mathrm{x}}$ let at_x be the smallest set of atoms such that (a) every atom in φ in which x occurs belongs to at_x, and (b) for every $v \in \mathrm{vars}(\mathrm{at}_x) \cap \bar{\mathrm{v}}$ with $\mathrm{idx}(v) \geq \mathrm{idx}(x)$ we have $\mathrm{at}(\mathcal{C}^\uparrow_v) \subseteq \mathrm{at}_x$. By at_0 we denote the set of all atoms that occur in φ but in none of the at_x with $x \in \bar{\mathrm{x}}$. Moreover, we use the notation $U_x := \mathrm{vars}(\mathrm{at}_x) \cap \bigcup_{i \geq \mathrm{idx}(x)} \bar{\mathrm{v}}_i$.

$\mathcal{G}_\varphi$

$\mathcal{C}^\uparrow_v$, $\mathrm{at}(\mathcal{C}^\uparrow_v)$

at_x, at_0

U_x

Claim I: If

(A) every atom in φ contains at most one variable from $\bar{\mathrm{x}}$, and

(B) for all distinct variables $x, x' \in \bar{\mathrm{x}}$ with $\mathrm{idx}(x) \leq \mathrm{idx}(x')$ and any variable $v \in \bigcup_{i \geq \mathrm{idx}(x)} \bar{\mathrm{v}}_i$ we have $v \notin \mathrm{vars}(\mathrm{at}_x) \cap \mathrm{vars}(\mathrm{at}_{x'})$,

then we observe the following properties:

(i) For all distinct $x, x' \in \bar{\mathrm{x}}$ we have $\mathrm{at}_x \cap \mathrm{at}_{x'} = \emptyset$.

(ii) For every $x \in \bar{\mathrm{x}}$ we have $\mathrm{vars}(\mathrm{at}_x) \cap \bar{\mathrm{x}} = \{x\}$.

(iii) For every $x \in \bar{\mathrm{x}}$ we have $U_x \cap \mathrm{vars}(\mathrm{at}_0) = \emptyset$.

(iv) For all distinct $x, x' \in \bar{\mathrm{x}}$ with $\mathrm{idx}(x) \leq \mathrm{idx}(x')$ we have $U_x \cap \mathrm{vars}(\mathrm{at}_{x'}) = \emptyset$.

Proof:

Ad (i): Suppose there are variables $x, x' \in \bar{\mathrm{x}}$ and there is some atom $A \in \mathrm{at}_x \cap \mathrm{at}_{x'}$. A must belong to $\mathrm{at}(\mathcal{C}^\uparrow_v)$ for some variable $v \in \bar{\mathrm{v}}$ with $\mathrm{idx}(v) \geq \mathrm{idx}(x)$ or $\mathrm{idx}(v) \geq \mathrm{idx}(x')$, since otherwise we would have $\{x, x'\} \subseteq \mathrm{vars}(A)$ which contradicts Condition (A). This in turn means that some variable $v' \in \bar{\mathrm{v}}$ occurs in A for which $\mathrm{idx}(v') \geq \mathrm{idx}(v)$. Hence, $v' \in \mathrm{vars}(A) \subseteq \big(\mathrm{vars}(\mathrm{at}_x) \cap \mathrm{vars}(\mathrm{at}_{x'})\big)$ with $\mathrm{idx}(v') \geq \mathrm{idx}(x)$ or $\mathrm{idx}(v') \geq \mathrm{idx}(x')$. This constitutes a contradiction to Condition (B).

Ad (ii): This is a direct consequence of (i) and the definition of at_x.

Ad (iii): Whenever $v \in U_x \subseteq \bigcup_{i \geq \mathrm{idx}(x)} \bar{\mathrm{v}}_i$ we have that $\mathrm{at}(\mathcal{C}^\uparrow_v) \subseteq \mathrm{at}_x$. Suppose there is some $v \in U_x \cap \mathrm{vars}(\mathrm{at}_0)$, i.e. there is some atom $A \in \mathrm{at}_0$ with $v \in \mathrm{vars}(A)$. Since $A \in \mathrm{at}(\mathcal{C}^\uparrow_v) \subseteq \mathrm{at}_x$, the definition of at_0 entails that A cannot occur in at_0.

Ad (iv): Whenever $v \in U_x \subseteq \bigcup_{i \geq \mathrm{idx}(x)} \bar{\mathrm{v}}_i$, we observe that $\mathrm{at}(\mathcal{C}^\uparrow_v) \subseteq \mathrm{at}_x$. Suppose there is some $v \in U_x \cap \mathrm{vars}(\mathrm{at}_{x'})$. Hence, there must be some atom $A \in \mathrm{at}_{x'}$ in which v occurs. But since A belongs to $\mathrm{at}(\mathcal{C}^\uparrow_v)$, we know that $A \in \mathrm{at}_x$. This contradicts (i). ◊

Claim II: The sentence φ belongs to GAF if and only if it satisfies Conditions (A) and (B) from Claim I.

Proof: Regarding the *if*-direction, we set $\mathrm{At}_0 := \mathrm{at}_0$ and $\mathrm{At}_x := \mathrm{at}_x$ for every $x \in \bar{\mathrm{x}}$. Moreover, every existential quantifier block $\exists \bar{\mathrm{v}}_k$ can be partitioned into $\exists \bar{\mathrm{y}}_k \exists \bar{\mathrm{u}}_k$ by setting $\bar{\mathrm{u}}_k := \bar{\mathrm{y}}_k \cap \bigcup_{x \in \bar{\mathrm{x}}} U_x$ and $\bar{\mathrm{y}}_k := \bar{\mathrm{v}}_k \setminus \bar{\mathrm{u}}_k$. Condition (a) of Definition 3.7.1 is satisfied due to the following observations. By definition of at_0 and the at_x we have $\mathrm{vars}(\mathrm{At}_0) \cap \bar{\mathrm{x}} = \emptyset$. By virtue of Claim I(iii) and the above partition of the $\bar{\mathrm{v}}_k$ into $\bar{\mathrm{y}}_k$ and $\bar{\mathrm{u}}_k$, we have $\bar{\mathrm{u}}_k \cap \mathrm{vars}(\mathrm{At}_0) = \emptyset$ for every k. Hence, $\mathrm{vars}(\mathrm{At}_0) \subseteq \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_n$. Condition (b) of Definition 3.7.1 follows because of the way we partition the $\bar{\mathrm{v}}_k$ into $\bar{\mathrm{y}}_k, \bar{\mathrm{u}}_k$. Any variable $v \in \bar{\mathrm{v}} \cap \mathrm{vars}(\mathrm{At}_x)$ with index $\mathrm{idx}(v) = k$ belongs to U_x if and only if $k \geq \mathrm{idx}(x)$. Hence, we have $v \in \bar{\mathrm{y}}_k$ if and only if $k < \mathrm{idx}(x)$, and we have $v \in \bar{\mathrm{u}}_k$ otherwise. Moreover, Claim I(ii) states that x is the only variable from $\bar{\mathrm{x}}$ that occurs in At_x. Condition (c) of Definition 3.7.1 follows immediately from Claim I(iv) and the fact that $\bar{\mathrm{u}}_1 \cup \ldots \cup \bar{\mathrm{u}}_n = \bigcup_{x \in \bar{\mathrm{x}}} U_x$.

Regarding the *only if*-direction we argue as follows. Condition (A) of Claim I is certainly satisfied, if Condition (b) of Definition 3.7.1 is met by φ. Consider any two variables $x, x' \in \bar{\mathrm{x}}$ with $\mathrm{idx}(x) \leq \mathrm{idx}(x')$ and let u be some variable in U_x. Because of $\mathrm{idx}(u) \geq \mathrm{idx}(x)$, Condition (b) of Definition 3.7.1 entails that $u \in \bar{\mathrm{u}}$. By Condition (c) of Definition 3.7.1, u cannot occur in both At_x and $\mathrm{At}_{x'}$. Since $U_x \subseteq \mathrm{vars}(\mathrm{at}_x) = \mathrm{vars}(\mathrm{At}_x)$, u cannot occur in $\mathrm{At}_{x'}$ and, hence, not in $\mathrm{at}_{x'}$ either. This entails that Condition (B) of Claim I is satisfied. $\Diamond$

By Claim II, Conditions (A) and (B) from Claim I together yield a criterion to decide whether φ belongs to GAF or not. It remains to argue that this criterion can be checked deterministically in polynomial time. It is straightforward to check Condition (A) in polynomial time. Hence, we concentrate on Condition (B). Given φ, the graph $\mathcal{G}_\varphi = \langle V, E\rangle$ can be constructed in time that is quadratic in $\|\varphi\|$. We observe $|V| = |\bar{\mathrm{v}}| \leq |\mathrm{len}(\varphi)|$ and $|E| \leq |\bar{\mathrm{v}}|^2 \leq \big(\mathrm{len}(\varphi)\big)^2$. Using efficient data structures, the upward closures $\mathcal{C}_v^{\uparrow}$ in $\mathcal{G}_\varphi$ and the sets $\mathrm{at}(\mathcal{C}_v^{\uparrow})$ and at_x can be computed in time that is polynomial in $\|\varphi\|$. The sum of the lengths of the atoms in all the at_x taken together is at most $|\bar{\mathrm{x}}| \cdot \mathrm{len}(\varphi) \leq \big(\mathrm{len}(\varphi)\big)^2$. Finally, the test whether we have

$$\mathrm{vars}(\mathrm{at}_x) \cap \mathrm{vars}(\mathrm{at}_{x'}) \cap \bigcup_{i \geq \mathrm{idx}(x)} \bar{\mathrm{v}}_i = \emptyset$$

for all distinct x, x' with $\mathrm{idx}(x) \leq \mathrm{idx}(x')$ can be done in time that is polynomial in $\|\varphi\|$. □

The next proposition confirms that GAF indeed extends the Ackermann fragment. Moreover, MFO is a proper subfragment of GAF. Since the sentence φ from Example 3.7.2 belongs to GAF but lies in neither of the other two fragments, it is immediately clear that GAF constitutes a proper syntactical extension of both.

Proposition 3.7.4. *GAF properly contains AF and MFO.*

Proof. Let $\varphi := \exists \bar{\mathrm{z}} \forall x \exists \bar{\mathrm{v}}.\, \psi$ be an AF sentence with quantifier-free ψ. Any atom in φ contains at most one universally quantified variable, namely x. Let At_x be the set of all atoms occurring in φ. If we conceive the variables in $\bar{\mathrm{v}}$ as "$\bar{\mathrm{u}}$-variables", then Condition (b) of Definition 3.7.1 is satisfied by At_x. The other two conditions, (a) and (c), are trivially satisfied. Consequently, φ belongs to GAF.

Let $\varphi' := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n.\, \psi'$ be an MFO sentence. For every $x \in \bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_n$ define At_x to be the set containing exactly the atoms in φ' that contain x. Let At_0 be the set of all atoms in φ' that do not belong to any At_x. Clearly, this partition of φ''s atoms meets all the conditions stated in Definition 3.7.1, if we conceive all the existentially quantified variables in φ' as "$\bar{\mathrm{y}}$-variables". Hence, φ' belongs to GAF. □

GAF-Sat

As for GBSR, we shall first discuss a syntactic route to decidability of the satisfiability for GAF (*GAF-Sat*) that is based on an effective equivalence-preserving translation from GAF into AF. A semantically-flavored approach based on model-checking games and the analysis of dependences between existentially and universally quantified variables shall be developed in Section 4.3. Both approaches will lead to small model properties for GAF and, as one consequence, yield upper bounds on the computational complexity of deciding GAF-Sat.

3.8 Translation of GAF into the Ackermann Fragment

The equivalence-preserving translation from GAF into AF proceeds in two stages. The first stage resembles an exhaustive unfolding process in the spirit of Lemma 2.0.3. Nestings of quantifiers in a given GAF sentence that bind separated sets of variables vanish in the course of this stage. This results in a sentence in which every subformula lies within the scope of at most one universal quantifier. Such sentences can easily be converted into a special syntactic form, which we shall call *GAF special form*. Then, in the second stage of the translation process, a sentence in GAF special form is transformed into an equivalent AF sentence.

The next lemma focuses on the first stage of the translation process.

Lemma 3.8.1. *Let $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \bar{\mathrm{u}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n \bar{\mathrm{u}}_n . \psi$ be any GAF sentence with quantifier-free ψ. We can effectively transform φ into an equivalent sentence φ' in standard form, in which every subformula lies within the scope of at most one universal quantifier. Moreover, all literals in φ' also occur in φ (modulo variable renaming).*

Proof. We assume that φ is in standard form. Let the sets At, $\bar{\mathrm{x}}$, $\bar{\mathrm{y}}$, $\bar{\mathrm{u}}$ be defined as in Definition 3.7.1. Let $\mathrm{At}_0, (\mathrm{At}_x)_{x \in \bar{\mathrm{x}}}$ be the partition of the set At described in Definition 3.7.1. Recall that φ is assumed to be in negation normal form. Let $\mathcal{L}_0, (\mathcal{L}_x)_{x \in \bar{\mathrm{x}}}$ be the corresponding partition of the set of literals occurring in φ. Hence, every At_x is exactly the set of atoms occurring in $\mathcal{L}_x$, and the same holds for At_0 and $\mathcal{L}_0$. Moreover, we use the notation $U_x := \mathrm{vars}(\mathcal{L}_x) \cap \bigcup_{i \geq \mathrm{idx}(x)} \bar{\mathrm{u}}_i$. $\mathrm{At}_0, \mathrm{At}_x$ $\mathcal{L}_0, \mathcal{L}_x$ U_x

Given any set V of variables, we write $\mathcal{L}(V)$ to address the set of all literals from φ that contain at least one variable from V. For every $x \in \bar{\mathrm{x}}$ we refine the set $\mathcal{L}_x$ into subsets $\mathcal{L}_{x,0}$, $\mathcal{L}_{x,\mathrm{idx}(x)}, \mathcal{L}_{x,\mathrm{idx}(x)+1}, \ldots, \mathcal{L}_{x,n}$: $\mathcal{L}(V)$ $\mathcal{L}_{x,k}$

$\mathcal{L}_{x,n} := \mathcal{L}_x \cap \mathcal{L}(\bar{\mathrm{u}}_n)$,
$\mathcal{L}_{x,k} := \big(\mathcal{L}_x \cap \mathcal{L}(\bar{\mathrm{u}}_k)\big) \setminus \bigcup_{\ell > k} \mathcal{L}_{x,\ell}$ for every k satisfying $\mathrm{idx}(x) \leq k < n$, and
$\mathcal{L}_{x,0} := \mathcal{L}_x \setminus \bigcup_{\ell \geq \mathrm{idx}(x)} \mathcal{L}_{x,\ell}$.

Similarly, we define $U_{x,k} := \mathrm{vars}(\mathcal{L}_{x,k}) \cap U_x$ for every k, $\mathrm{idx}(x) \leq k \leq n$. $U_{x,k}$

Then, we observe the following (cf. Claim I in the proof of Theorem 3.7.3):

(I) For all distinct $x, x' \in \bar{\mathrm{x}}$ we have $\mathcal{L}_x \cap \mathcal{L}_{x'} = \emptyset$.

(II) For all distinct $x, x' \in \bar{\mathrm{x}}$ with $\mathrm{idx}(x) \leq \mathrm{idx}(x')$ we have $U_x \cap \mathrm{vars}(\mathcal{L}_{x'}) = \emptyset$.

(III) For every k we have $\mathcal{L}_{x,k} \subseteq \mathcal{L}_x$.

(IV) For all distinct k, ℓ we have $\mathcal{L}_{x,k} \cap \mathcal{L}_{x,\ell} = \emptyset$.

(V) For every $k \geq \mathrm{idx}(x)$ we have $\mathrm{vars}(\mathcal{L}_{x,k}) \subseteq \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_{\mathrm{idx}(x)-1} \cup \{x\} \cup \bar{\mathrm{u}}_{\mathrm{idx}(x)} \cup \ldots \cup \bar{\mathrm{u}}_k$.

(VI) We have $\mathrm{vars}(\mathcal{L}_{x,0}) \subseteq \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_{\mathrm{idx}(x)-1} \cup \{x\}$.

After having fixed notation, we proceed along similar lines as in the proof of Lemma 3.5.2, i.e. we perform syntactic transformations based on the axioms of Boolean algebra and ordinary quantifier shifting (cf. Lemma 1.0.1). Once more, this will not change the set of literals occurring in the intermediate steps (modulo variable renaming), since we start from a formula in negation normal form restricted to the connectives $\wedge, \vee, \neg$. Analogous to the proof of Lemma 3.5.2, we (re-)transform parts of φ repeatedly into a disjunction of conjunctions (or a conjunction of disjunctions) of subformulas which we treat as indivisible units. The literals and indivisible units in the respective conjunctions (disjunctions) will be grouped in accordance with the sets $\mathcal{L}_0, \mathcal{L}_x$ and $\mathcal{L}_{x,\mathrm{idx}(x)}, \ldots, \mathcal{L}_{x,n}$, where needed. For this purpose, it is important to keep in mind that (I) and the definition of $\mathcal{L}_0$ entail that $\mathcal{L}_0$ together with the sets $\mathcal{L}_x$ partition the set of all literals occurring in φ. Moreover, every $\mathcal{L}_x$ is partitioned by the sets $\mathcal{L}_{x,0}, \mathcal{L}_{x,\mathrm{idx}(x)}, \ldots, \mathcal{L}_{x,n}$, by virtue of (III), (IV), and the definition of $\mathcal{L}_{x,0}$.

Let us elaborate on transformation process: we first describe it and then it is presented formally. At the beginning, we transform ψ into a disjunction of conjunctions of literals $\bigvee_i \psi_i$. Then, we rewrite every ψ_i into $\chi^{(1)}_{i,0} \wedge \bigwedge_{k=1}^{n} \bigwedge_{x \in \bar{\mathrm{x}}_k} \big(\chi^{(1)}_{i,x,0} \wedge \bigwedge_{j=k}^{n} \chi^{(1)}_{i,x,j}\big)$, where $\chi^{(1)}_{i,0}$ and the $\chi^{(1)}_{i,x,j}$ are conjunctions of literals. $\chi^{(1)}_{i,0}$ comprises all literals in ψ_i which belong to $\mathcal{L}_0$, while for every j the literals which belong to $\mathcal{L}_{x,j}$ are grouped into $\chi^{(1)}_{i,x,j}$, respectively. By (V) and (VI), we know that $\mathrm{vars}(\chi^{(1)}_{i,x,0}) \subseteq \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_{\mathrm{idx}(x)-1} \cup \{x\}$ and $\mathrm{vars}(\chi^{(1)}_{i,x,j}) \subseteq \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_{\mathrm{idx}(x)-1} \cup \{x\} \cup \bar{\mathrm{u}}_{\mathrm{idx}(x)} \cup \ldots \cup \bar{\mathrm{u}}_j$ for $j > 0$. Moreover, the definition of $\mathcal{L}_0$ entails $\mathrm{vars}(\chi^{(1)}_{i,0}) \subseteq \bar{\mathrm{y}}$.

At this point, we shift the existential quantifier block $\exists \bar{\mathrm{y}}_n \bar{\mathrm{u}}_n$ inwards. By (VI), we have $\mathrm{vars}(\mathcal{L}_{x,0}) \cap (\bar{\mathrm{y}}_n \cup \bar{\mathrm{u}}_n) = \emptyset$. Therefore, the subformulas $\chi^{(1)}_{i,x,0}$ contain neither variables from $\bar{\mathrm{y}}_n$ nor from $\bar{\mathrm{u}}_n$. Similarly, due to (V), the $\chi^{(1)}_{i,x,j}$ with $0 < j < n$ do not contain any variables from $\bar{\mathrm{y}}_n$ or from $\bar{\mathrm{u}}_n$. Consequently, one part of the quantifier block $\exists \bar{\mathrm{y}}_n \bar{\mathrm{u}}_n$, namely $\exists\big(\bar{\mathrm{y}}_n \cap \mathrm{vars}(\chi^{(1)}_{i,0})\big)$, binds

variables in $\chi^{(1)}_{i,0}$ (for convenience, we still write the full $\exists\bar{\mathrm{y}}_n$, which does not affect semantics), and another — disjoint — part, namely $\exists\big(\bar{\mathrm{u}}_n \cap U_{x,n}\big)$, binds variables in $\chi^{(1)}_{i,x,n}$. In addition, (II) ensures that $\bar{\mathrm{u}} \cap U_{x,n}$ is disjoint from $\bar{\mathrm{u}} \cap U_{x',n}$ for distinct x, x'. The thus obtained sentence φ'' has the form

$$\forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\bar{\mathrm{u}}_1 \ldots \forall\bar{\mathrm{x}}_n.\ \bigvee_i \big(\exists\bar{\mathrm{y}}_n.\, \chi^{(1)}_{i,0}\big) \wedge \bigwedge_{k=1}^{n} \bigwedge_{x\in\bar{\mathrm{x}}_k} \big(\chi^{(1)}_{i,x,0} \wedge \bigwedge_{j=k}^{n-1} \chi^{(1)}_{i,x,j} \wedge \exists(\bar{\mathrm{u}}_n \cap U_{x,n}).\, \chi^{(1)}_{i,x,n}\big)\ .$$

In the rest of the transformation process we treat the subformulas $\big(\exists\bar{\mathrm{y}}_n.\, \chi^{(1)}_{i,0}\big)$ and $\big(\exists(\bar{\mathrm{u}}_n \cap U_{x,n}).\, \chi^{(1)}_{i,x,n}\big)$ as indivisible units.

Next, we transform the big disjunction in φ'' into a conjunction of disjunctions $\bigwedge_i \psi'_i$, rewrite the disjunctions ψ'_i into subformulas $\eta^{(1)}_{i,0} \vee \big(\bigvee_{k=1}^{n-1}\bigvee_{x\in\bar{\mathrm{x}}_k} \eta^{(1)}_{i,x}\big) \vee \bigvee_{x\in\bar{\mathrm{x}}_n} \eta^{(1)}_{i,x}$, similarly to what we have done above, but this time grouped in accordance with the more coarse-grained sets $\mathcal{L}_0$ and $\mathcal{L}_x$. Having done the regrouping, we shift the universal quantifier block $\forall\bar{\mathrm{x}}_n$ inwards. The resulting formula has the shape $\forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\bar{\mathrm{u}}_1 \ldots \forall\bar{\mathrm{x}}_{n-1}\exists\bar{\mathrm{y}}_{n-1}\bar{\mathrm{u}}_{n-1}.\, \bigwedge_i \eta^{(1)}_{i,0} \vee \big(\bigvee_{k=1}^{n-1}\bigvee_{x\in\bar{\mathrm{x}}_k} \eta^{(1)}_{i,x}\big) \vee \bigvee_{x\in\bar{\mathrm{x}}_n} \forall x.\, \eta^{(1)}_{i,x}$. From this point on we treat the subformulas $\big(\forall x.\, \eta^{(1)}_{i,x}\big)$ as indivisible units as well. Moreover, we shall group them under the conjunctions $\chi^{(\ell)}_{i,0}$ or $\eta^{(\ell)}_{i,0}$, $\ell \geq 2$, respectively, since they do not contain any free occurrences of universally quantified variables $x \in \bar{\mathrm{x}}$ anymore. This is not only convenient but also necessary, because a subformula $\big(\forall x.\, \eta^{(1)}_{i,x}\big)$ may share free variables $y \in \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_{\mathrm{idx}(x)-1}$ with the subformula $\eta^{(1)}_{i,0}$. Hence, when some quantifier $\exists y$ is shifted inwards later on, both $\big(\forall x.\, \eta^{(1)}_{i,x}\big)$ and some literals in $\eta^{(1)}_{i,0}$ might have to remain within the scope of $\exists y$.

We reiterate the described process until all the quantifiers have been shifted inwards in the outlined way. There is one more peculiarity to mention. At later stages of the transformation subformulas of the form $\chi^{(\ell)}_{i,x,j} \wedge \ldots \wedge \chi^{(\ell)}_{i,x,n}$ may appear in which the constituents $\chi^{(\ell)}_{i,x,j'}$ may share variables $u \in \bar{\mathrm{u}}_j$, for instance. For the sake of readability, we abbreviate such subformulas by $\chi^{(\ell)}_{i,x,\geq j}$ and similar notations. Emerging subformulas $\big(\exists(\bar{\mathrm{u}}_\ell \cap U_{x,\ell}).\, \chi^{(\ell)}_{i,x,\geq j}\big)$ will be treated as indivisible units.

$$\begin{aligned}
&\forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\bar{\mathrm{u}}_1 \ldots \forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n\bar{\mathrm{u}}_n.\, \psi \\
&\quad\models\!\models\ \forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\bar{\mathrm{u}}_1 \ldots \forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n\bar{\mathrm{u}}_n.\, \bigvee_i \chi^{(1)}_{i,0}(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n) \\
&\qquad\qquad \wedge \bigwedge_{k=1}^{n} \bigwedge_{x\in\bar{\mathrm{x}}_k} \Big(\chi^{(1)}_{i,x,0}(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_{k-1}, x) \\
&\qquad\qquad\qquad \wedge \bigwedge_{j=k}^{n} \chi^{(1)}_{i,x,j}(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_{k-1}, x, \bar{\mathrm{u}}_k, \ldots, \bar{\mathrm{u}}_j)\Big) \\
&\quad\models\!\models\ \forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\bar{\mathrm{u}}_1 \ldots \forall\bar{\mathrm{x}}_n.\, \bigvee_i \Big(\exists\bar{\mathrm{y}}_n.\, \chi^{(1)}_{i,0}(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n)\Big) \\
&\qquad\qquad \wedge \bigwedge_{k=1}^{n} \bigwedge_{x\in\bar{\mathrm{x}}_k} \Big(\chi^{(1)}_{i,x,0}(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_{k-1}, x) \\
&\qquad\qquad\qquad \wedge \bigwedge_{j=k}^{n-1} \chi^{(1)}_{i,x,j}(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_{k-1}, x, \bar{\mathrm{u}}_k, \ldots, \bar{\mathrm{u}}_j) \\
&\qquad\qquad\qquad \wedge \exists(\bar{\mathrm{u}}_n \cap U_{x,n}).\, \chi^{(1)}_{i,x,n}(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_{k-1}, x, \bar{\mathrm{u}}_k, \ldots, \bar{\mathrm{u}}_n)\Big)
\end{aligned}$$

$$\models\ \forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\bar{\mathrm{u}}_1\ldots\forall\bar{\mathrm{x}}_n.\bigwedge_i\eta^{(1)}_{i,0}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_{n-1})$$
$$\vee\bigvee_{k=1}^{n-1}\bigvee_{x\in\bar{\mathrm{x}}_k}\eta^{(1)}_{i,x}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_{k-1},x,\bar{\mathrm{u}}_k,\ldots,\bar{\mathrm{u}}_{n-1})$$
$$\vee\bigvee_{x\in\bar{\mathrm{x}}_n}\eta^{(1)}_{i,x}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_{n-1},x)$$

$$\models\ \forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\bar{\mathrm{u}}_1\ldots\forall\bar{\mathrm{x}}_{n-1}\exists\bar{\mathrm{y}}_{n-1}\bar{\mathrm{u}}_{n-1}.\bigwedge_i\eta^{(1)}_{i,0}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_{n-1})$$
$$\vee\bigvee_{k=1}^{n-1}\bigvee_{x\in\bar{\mathrm{x}}_k}\eta^{(1)}_{i,x}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_{k-1},x,\bar{\mathrm{u}}_k,\ldots,\bar{\mathrm{u}}_{n-1})$$
$$\vee\bigvee_{x\in\bar{\mathrm{x}}_n}\Big(\forall x.\,\eta^{(1)}_{i,x}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_{n-1},x)\Big)$$

$$\models\ \forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\bar{\mathrm{u}}_1\ldots\forall\bar{\mathrm{x}}_{n-1}\exists\bar{\mathrm{y}}_{n-1}\bar{\mathrm{u}}_{n-1}.$$
$$\bigvee_i\chi^{(2)}_{i,0}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_{n-1})$$
$$\wedge\bigwedge_{k=1}^{n-1}\bigwedge_{x\in\bar{\mathrm{x}}_k}\Big(\chi^{(2)}_{i,x,0}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_{k-1},x)\wedge\bigwedge_{j=k}^{n-2}\chi^{(2)}_{i,x,j}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_{k-1},x,\bar{\mathrm{u}}_k,\ldots,\bar{\mathrm{u}}_j)$$
$$\wedge\chi^{(2)}_{i,x,\geq n-1}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_{k-1},x,\bar{\mathrm{u}}_k,\ldots,\bar{\mathrm{u}}_{n-1})\Big)$$

$$\models\ \forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\bar{\mathrm{u}}_1\ldots\forall\bar{\mathrm{x}}_{n-1}.$$
$$\bigvee_i\Big(\exists\bar{\mathrm{y}}_{n-1}.\,\chi^{(2)}_{i,0}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_{n-1})\Big)$$
$$\wedge\bigwedge_{k=1}^{n-1}\bigwedge_{x\in\bar{\mathrm{x}}_k}\Big(\chi^{(2)}_{i,x,0}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_{k-1},x)\wedge\bigwedge_{j=k}^{n-2}\chi^{(2)}_{i,x,j}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_{k-1},x,\bar{\mathrm{u}}_k,\ldots,\bar{\mathrm{u}}_j)$$
$$\wedge\Big(\exists(\bar{\mathrm{u}}_{n-1}\cap U_{x,n-1}).\,\chi^{(2)}_{i,x,\geq n-1}(\bar{\mathrm{y}}_1,\ldots,\bar{\mathrm{y}}_{k-1},x,\bar{\mathrm{u}}_k,\ldots,\bar{\mathrm{u}}_{n-1})\Big)\Big)$$

$$\vdots$$

$$\models\ \forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\bar{\mathrm{u}}_1.\bigvee_i\chi^{(n)}_{i,0}(\bar{\mathrm{y}}_1)\ \wedge\bigwedge_{x\in\bar{\mathrm{x}}_1}\chi^{(n)}_{i,x,0}(x)\wedge\chi^{(n)}_{i,x,\geq 1}(x,\bar{\mathrm{u}}_1)$$

$$\models\ \forall\bar{\mathrm{x}}_1.\bigvee_i\Big(\exists\bar{\mathrm{y}}_1.\,\chi^{(n)}_{i,0}(\bar{\mathrm{y}}_1)\Big)\ \wedge\bigwedge_{x\in\bar{\mathrm{x}}_1}\chi^{(n)}_{i,x,0}(x)\wedge\big(\exists(\bar{\mathrm{u}}_1\cap U_{x,1}).\,\chi^{(n)}_{i,x,\geq 1}(x,\bar{\mathrm{u}}_1)\big)$$

$$\models\ \forall\bar{\mathrm{x}}_1.\bigwedge_i\eta^{(n)}_{i,0}()\ \vee\bigvee_{x\in\bar{\mathrm{x}}_1}\eta^{(n)}_{i,x}(x)$$

$$\models\ \bigwedge_i\eta^{(n)}_{i,0}()\ \vee\bigvee_{x\in\bar{\mathrm{x}}_1}\forall x.\,\eta^{(n)}_{i,x}(x)$$

The final result of this transformation is the sought φ'. Every time a universal quantifier block $\forall\bar{\mathrm{x}}_j$ is shifted inwards at the ℓ-th stage, all the subformulas which contain universal quantifiers already are grouped into $\eta^{(\ell)}_{i,0}$. Due to the disjointness properties of the $\mathcal{L}_{x,k}$ and the $U_{x,k}$, it is guaranteed that no $\eta^{(\ell)}_{i,0}$ contains a free occurrence of any $x\in\bar{\mathrm{x}}$ (details have been elaborated above). Consequently, in the final result φ' we do not have any nested occurrences of universal quantifiers.

We restore the property that no two quantifiers in φ' bind the same variables by appropriately renaming bound variables in φ'. □

GAF special form

The sentence φ' whose existence is stipulated in Lemma 3.8.1 can easily be further transformed into a particular shape to which we shall refer as *GAF special form*:

$$\exists \bar{z}.\, \bigwedge_i \Big(\bigvee_j \forall x_{i,j} \exists \bar{y}_{i,j}.\, \chi_{i,j}(\bar{z}, x_{i,j}, \bar{y}_{i,j}) \Big) \vee \eta_i(\bar{z})$$

where the $\chi_{i,j}$ and the η_i are quantifier free.

Lemma 3.8.2 (GAF special form)**.** *If φ belongs to GAF, then we can effectively construct an equivalent sentence of the form*

$$\exists \bar{z}.\, \textstyle\bigwedge_i \big(\bigvee_j \forall x_{i,j} \exists \bar{y}_{i,j}.\, \chi_{i,j}(\bar{z}, x_{i,j}, \bar{y}_{i,j}) \big) \vee \eta_i(\bar{z}),$$

where the $\chi_{i,j}$ and the η_i are quantifier free.

Proof. By Lemma 3.8.1, we can effectively construct a sentence φ' in standard form that is equivalent to φ and that does not contain any nested occurrences of universal quantifiers. We construct φ'' from φ' as follows. First, we shift all existential quantifiers in φ' that do not lie within the scope of any universal quantifier to the front of the formula. In the resulting sentence $\exists \bar{z}.\, \psi'$ every existential quantifier in ψ' lies within the scope of exactly one universal quantifier. We treat every subformula of the form $\forall x.\, \chi$ in ψ' as indivisible unit while transforming ψ' into an equivalent conjunction of disjunctions of literals and such indivisible units. The resulting formula can be brought into the desired shape by shifting existential quantifiers that lie in the scope of a universal quantifier outwards until they form an existential quantifier block directly right of the corresponding universal quantifier. □

It is interesting to note that a sentence in GAF special form is not merely a Boolean combination of Ackermann sentences. The difference is that distinct subformulas $\forall x \exists \bar{y}.\, \chi$ and $\forall x' \exists \bar{y}'.\, \chi'$ may share existentially quantified variables. However, one can show that every such sentence is indeed equivalent to some Ackermann sentence. Therefore, every GAF sentence is equivalent to an Ackermann sentence. Before we make this claim precise (cf. Lemma 3.8.4), we develop an auxiliary result that we will reuse later.

Lemma 3.8.3. *Let ψ be a first-order formula of the form $\psi := \bigvee_j \forall \bar{x}\, \exists \bar{y}.\, \chi_j(\bar{z}, \bar{x}, \bar{y})$ with quantifier-free subformulas $\chi_j(\bar{z}, \bar{x}, \bar{y})$. Then, ψ is equivalent to some formula ψ' of the form*

$$\psi' := \exists \bar{v}_1 \ldots \bar{v}_q \exists \bar{y}_1 \ldots \bar{y}_q.\, \Big(\bigvee_j \bigwedge_{k=1}^{q} \chi_j(\bar{z}, \bar{v}_k, \bar{y}_k) \Big) \wedge \forall \bar{x} \exists \bar{y}.\, \bigvee_{k=1}^{q} \bigwedge_{A \in \mathrm{At}} \big(A(\bar{z}, \bar{x}, \bar{y}) \leftrightarrow A(\bar{z}, \bar{v}_k, \bar{y}_k) \big)\ ,$$

where At *denotes the set of all atoms occurring in ψ and $q := 2^{|\mathrm{At}|}$. In addition, we have $|\bar{v}_k| = |\bar{x}|$ for every k and $|\bar{y}_\ell| = |\bar{y}|$ for every ℓ.*

D_S

Proof. We first prove $\psi \models \psi'$. Let $\mathcal{A}$ be any structure, β any variable assignment, and j any index such that $\mathcal{A}, \beta \models \forall \bar{x}\, \exists \bar{y}.\, \chi_j(\bar{z}, \bar{x}, \bar{y})$. For every set $S \subseteq \mathrm{At}$ we define

$$\mathsf{D}_S := \big\{ \langle \bar{\mathsf{a}}, \bar{\mathsf{c}} \rangle \in \mathsf{D}^{|\bar{x}|+|\bar{y}|} \;\big|\; \text{for every atom } A \in \mathrm{At} \text{ we have } \mathcal{A}, \beta[\bar{x} \mapsto \bar{\mathsf{a}}, \bar{y} \mapsto \bar{\mathsf{c}}\,] \models A(\bar{z}, \bar{x}, \bar{y}) \text{ if and only if } A \in S \big\}\ .$$

We write $S \models \chi_j(\bar{z}, \bar{x}, \bar{y})$ if D_S is nonempty and if we have $\mathcal{A}, \beta[\bar{x} \mapsto \bar{\mathsf{a}}, \bar{y} \mapsto \bar{\mathsf{c}}\,] \models \chi_j(\bar{z}, \bar{x}, \bar{y})$ for every tuple $\langle \bar{\mathsf{a}}, \bar{\mathsf{c}} \rangle$ in D_S. Let $S_1, \ldots, S_r$ be an enumeration of all the sets S_k with $S_k \models \chi_j(\bar{z}, \bar{x}, \bar{y})$. Notice that $1 \leq r \leq q$. Let $\langle \bar{\mathsf{b}}_1, \bar{\mathsf{c}}_1 \rangle, \ldots, \langle \bar{\mathsf{b}}_r, \bar{\mathsf{c}}_r \rangle$ be some sequence with $\langle \bar{\mathsf{b}}_k, \bar{\mathsf{c}}_k \rangle \in \mathsf{D}_{S_k}$ for every k. Then, for every k the assumption $S_k \models \chi_j(\bar{z}, \bar{x}, \bar{y})$ entails $\mathcal{A}, \beta[\bar{x} \mapsto \bar{\mathsf{b}}_k, \bar{y} \mapsto \bar{\mathsf{c}}_k] \models \chi_j(\bar{z}, \bar{x}, \bar{y})$. Hence,

$$\mathcal{A}, \beta[\bar{v}_1 \mapsto \bar{\mathsf{b}}_1, \ldots, \bar{v}_r \mapsto \bar{\mathsf{b}}_r, \bar{v}_{r+1} \mapsto \bar{\mathsf{b}}_1, \ldots, \bar{v}_q \mapsto \bar{\mathsf{b}}_1] \models \exists \bar{y}_1 \ldots \exists \bar{y}_q.\, \bigwedge_{k=1}^{q} \chi_j(\bar{z}, \bar{v}_k, \bar{y}_k)\ . \tag{3.4}$$

Let $\bar{\mathsf{a}} \in \mathsf{D}^{|\bar{\mathrm{x}}|}$ be any tuple of length $|\bar{\mathrm{x}}|$. Because of $\mathcal{A}, \beta \models \forall \bar{\mathrm{x}}\, \exists \bar{\mathrm{y}}.\, \chi_j(\bar{\mathrm{z}}, \bar{\mathrm{x}}, \bar{\mathrm{y}})$, there is some S_k, $1 \leq k \leq r$, and some tuple $\bar{\mathsf{c}} \in \mathsf{D}^{|\bar{\mathrm{y}}|}$ such that $\langle \bar{\mathsf{a}}, \bar{\mathsf{c}} \rangle \in \mathsf{D}_{S_k}$ and $S_k \models \chi_j(\bar{\mathrm{z}}, \bar{\mathrm{x}}, \bar{\mathrm{y}})$. Therefore, we get the following for $\langle \bar{\mathsf{b}}_k, \bar{\mathsf{c}}_k \rangle$:

$$\mathcal{A}, \beta[\bar{\mathrm{x}} \mapsto \bar{\mathsf{a}}, \bar{\mathrm{v}}_k \mapsto \bar{\mathsf{b}}_k, \bar{\mathrm{y}}_k \mapsto \bar{\mathsf{c}}_k] \models \exists \bar{\mathrm{y}}.\ \bigwedge_{A \in \mathrm{At}} \big(A(\bar{\mathrm{z}}, \bar{\mathrm{x}}, \bar{\mathrm{y}}) \leftrightarrow A(\bar{\mathrm{z}}, \bar{\mathrm{v}}_k, \bar{\mathrm{y}}_k) \big) \ . \tag{3.5}$$

Put together, (3.4) and (3.5) entail

$$\begin{aligned} &\mathcal{A}, \beta[\bar{\mathrm{v}}_1 \mapsto \bar{\mathsf{b}}_1, \ldots, \bar{\mathrm{v}}_r \mapsto \bar{\mathsf{b}}_r, \bar{\mathrm{v}}_{r+1} \mapsto \bar{\mathsf{b}}_1, \ldots, \bar{\mathrm{v}}_q \mapsto \bar{\mathsf{b}}_1] \models \\ &\qquad \exists \bar{\mathrm{y}}_1 \ldots \exists \bar{\mathrm{y}}_q. \Big(\bigvee_j \bigwedge_{k=1}^{q} \chi_j(\bar{\mathrm{z}}, \bar{\mathrm{v}}_k, \bar{\mathrm{y}}_k) \Big) \ \wedge\ \forall \bar{\mathrm{x}}\, \exists \bar{\mathrm{y}}.\ \bigvee_{k=1}^{q} \bigwedge_{A \in \mathrm{At}} \big(A(\bar{\mathrm{z}}, \bar{\mathrm{x}}, \bar{\mathrm{y}}) \leftrightarrow A(\bar{\mathrm{z}}, \bar{\mathrm{v}}_k, \bar{\mathrm{y}}_k) \big) \ . \end{aligned}$$

This proves $\mathcal{A}, \beta \models \psi'$. Hence, we have shown that $\mathcal{A}, \beta \models \psi$ implies $\mathcal{A}, \beta \models \psi'$.

Next, we show $\psi' \models \psi$. Let $\mathcal{A}$ be a structure, let β be a variable assignment, and let $\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_q, \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_q$ be tuples such that

$$\begin{aligned} &\mathcal{A}, \beta[\bar{\mathrm{v}}_1 \mapsto \bar{\mathsf{b}}_1, \ldots, \bar{\mathrm{v}}_q \mapsto \bar{\mathsf{b}}_q, \bar{\mathrm{y}}_1 \mapsto \bar{\mathsf{c}}_1, \ldots, \bar{\mathrm{y}}_q \mapsto \bar{\mathsf{c}}_q] \models \\ &\qquad \Big(\bigvee_j \bigwedge_{k=1}^{q} \chi_j(\bar{\mathrm{z}}, \bar{\mathrm{v}}_k, \bar{\mathrm{y}}_k) \Big) \ \wedge\ \forall \bar{\mathrm{x}}\, \exists \bar{\mathrm{y}}.\ \bigvee_{k=1}^{q} \bigwedge_{A \in \mathrm{At}} \big(A(\bar{\mathrm{z}}, \bar{\mathrm{x}}, \bar{\mathrm{y}}) \leftrightarrow A(\bar{\mathrm{z}}, \bar{\mathrm{v}}_k, \bar{\mathrm{y}}_k) \big) \ . \end{aligned} \tag{3.6}$$

Then, there is some index j such that

$$\mathcal{A}, \beta[\bar{\mathrm{v}}_1 \mapsto \bar{\mathsf{b}}_1, \ldots, \bar{\mathrm{v}}_q \mapsto \bar{\mathsf{b}}_q, \bar{\mathrm{y}}_1 \mapsto \bar{\mathsf{c}}_1, \ldots, \bar{\mathrm{y}}_q \mapsto \bar{\mathsf{c}}_q] \models \bigwedge_{k=1}^{q} \chi_j(\bar{\mathrm{z}}, \bar{\mathrm{v}}_k, \bar{\mathrm{y}}_k) \ .$$

Let $\mathsf{D}_1, \ldots, \mathsf{D}_q$ be sets defined such that D_k

$$\begin{aligned} \mathsf{D}_k := \big\{ \bar{\mathsf{a}} \in \mathsf{D}^{|\bar{\mathrm{x}}|} \mid &\text{ there is some tuple } \bar{\mathsf{c}} \in \mathsf{D}^{|\bar{\mathrm{y}}|} \text{ such that for every atom } A \in \mathrm{At} \\ &\text{we have } \mathcal{A}, \beta[\bar{\mathrm{x}} \mapsto \bar{\mathsf{a}}, \bar{\mathrm{y}} \mapsto \bar{\mathsf{c}}\,] \models A(\bar{\mathrm{z}}, \bar{\mathrm{x}}, \bar{\mathrm{y}}) \text{ if and only if} \\ &\mathcal{A}, \beta[\bar{\mathrm{v}}_k \mapsto \bar{\mathsf{b}}_k, \bar{\mathrm{y}}_k \mapsto \bar{\mathsf{c}}_k] \models A(\bar{\mathrm{z}}, \bar{\mathrm{v}}_k, \bar{\mathrm{y}}_k) \big\} \ . \end{aligned}$$

Note that the sets D_k are all nonempty but not necessarily pairwise disjoint. Then, because of Assumption (3.6), for every $\bar{\mathsf{a}} \in \mathsf{D}^{|\bar{\mathrm{x}}|}$ there is some k, $1 \leq k \leq q$, such that $\bar{\mathsf{a}} \in \mathsf{D}_k$. Because of $\mathcal{A}, \beta[\bar{\mathrm{v}}_k \mapsto \bar{\mathsf{b}}_k, \bar{\mathrm{y}}_k \mapsto \bar{\mathsf{c}}_k] \models \chi_j(\bar{\mathrm{z}}, \bar{\mathrm{v}}_k, \bar{\mathrm{y}}_k)$, we therefore have $\mathcal{A}, \beta[\bar{\mathrm{x}} \mapsto \bar{\mathsf{a}}, \bar{\mathrm{y}} \mapsto \bar{\mathsf{c}}\,] \models \chi_j(\bar{\mathrm{z}}, \bar{\mathrm{x}}, \bar{\mathrm{y}})$ for some tuple $\bar{\mathsf{c}} \in \mathsf{D}^{|\bar{\mathrm{y}}|}$. In other words, we have $\mathcal{A}, \beta \models \forall \bar{\mathrm{x}}\, \exists \bar{\mathrm{y}}.\, \chi_j(\bar{\mathrm{z}}, \bar{\mathrm{x}}, \bar{\mathrm{y}})$ which entails $\mathcal{A}, \beta \models \psi$. Hence, we have shown that $\mathcal{A}, \beta \models \psi'$ implies $\mathcal{A}, \beta \models \psi$. □

Lemma 3.8.3 is essential for the second stage in the transformation process between GAF and AF. With this tool at hand, the following lemma is now easy to prove.

Lemma 3.8.4. *For every GAF sentence φ we can effectively construct an equivalent sentence φ' over the same vocabulary that has the shape $\exists \bar{\mathrm{v}} \forall x \exists \bar{\mathrm{w}}.\, \psi$ with quantifier-free ψ.*

Proof sketch. By virtue of Lemma 3.8.2, we can transform φ into an equivalent sentence φ'' in GAF special form, i.e. $\varphi'' = \exists \bar{z}.\, \bigwedge_i \big(\bigvee_j \forall x_{i,j} \exists \bar{\mathrm{y}}_{i,j}.\, \chi_{i,j}(\bar{z}, x_{i,j}, \bar{\mathrm{y}}_{i,j}) \big) \vee \eta_i(\bar{z})$, where the $\chi_{i,j}$ and the η_i are quantifier free. Consider any subformula of the form $\psi' := \bigvee_j \forall x\, \exists \bar{\mathrm{y}}.\, \chi_j(\bar{\mathrm{z}}, x, \bar{\mathrm{y}})$, possibly containing free variables from $\bar{\mathrm{z}}$. By virtue of Lemma 3.8.3, ψ' is equivalent to some formula of the form $\exists \bar{\mathrm{v}}' \bar{\mathrm{y}}'.\, \chi'(\bar{\mathrm{z}}, \bar{\mathrm{v}}', \bar{\mathrm{y}}') \wedge \forall x \exists \bar{\mathrm{y}}.\, \chi''(\bar{\mathrm{z}}, x, \bar{\mathrm{y}}, \bar{\mathrm{v}}', \bar{\mathrm{y}}')$ with quantifier-free χ', χ''. Hence, φ'' is equivalent to some sentence that, after shifting some quantifiers outwards, is of the form $\exists \bar{z}.\, \bigwedge_i \big(\exists \bar{\mathrm{u}}_i \forall x_i \exists \bar{\mathrm{w}}_i.\, \psi_i''(\bar{\mathrm{z}}, \bar{\mathrm{u}}_i, x_i, \bar{\mathrm{w}}_i) \big) \vee \eta_i(\bar{\mathrm{z}})$, where the ψ_i'' and the η_i are quantifier free. A prenex version of this sentence yields the sought φ', since the universal quantifiers distribute over the topmost conjunction. □

Notice that the proofs of Lemmas 3.8.1 to 3.8.4 still work in the presence of the equality predicate or function symbols. Therefore, we obtain the following result.

Theorem 3.8.5. *Every GAF sentence φ is equivalent to some AF sentence ψ. Moreover, we get the following results for relaxed restrictions on the syntax.*

(a) Every GAF sentence with equality is equivalent to some AF sentence with equality.

(b) Every GAF sentence with arbitrary function symbols and without equality is equivalent to some Gurevich–Maslov–Orevkov sentence ($\exists^*\forall\exists^*$-sentences with arbitrary function symbols, cf. page 24).

(c) Every GAF sentence with equality and with a single unary function symbol is equivalent to some Shelah sentence ($\exists^*\forall\exists^*$-sentences with equality and a single unary function symbol, cf. page 24).

In addition, constant symbols are admissible in all of the above cases.

Since AF possesses the finite model property, so does GAF, even in the first two syntactically extended cases mentioned in Theorem 3.8.5. On the other hand, it is known that the Shelah fragment contains infinity axioms. One example is the sentence $\forall x\exists y.\, f(f(y)) \approx f(x) \wedge f(y) \not\approx x$ ([BGG97], proof of Proposition 6.5.5). Still, the satisfiability problem for the Shelah fragment is known to be decidable (cf. [BGG97], Section 7.3). Therefore, we get the following positive results regarding the decidability of GAF-Sat.

Corollary 3.8.6. *GAF-Sat is decidable, even in the syntactically more liberal cases given in Theorem 3.8.5. The syntactic extensions of GAF described in items (a) and (b) of Theorem 3.8.5 enjoy the finite model property.*

Remark 3.8.7. *Every sentence from the Löb–Gurevich fragment (monadic first-order sentences with constant symbols and unary function symbols but without equality, cf. page 23) falls into the syntactic category of GAF when we in addition allow unary function symbols. By Lemma 3.8.4, every such sentence is equivalent to some $\exists^*\forall\exists^*$-sentence over the same vocabulary. The latter kind of sentences constitutes a subclass of the Gurevich–Maslov–Orevkov fragment. Hence, Lemmas 3.8.1 to 3.8.4 establish a reduction of the satisfiability problem for sentences from the Löb–Gurevich fragment to the satisfiability problem for the Gurevich–Maslov–Orevkov fragment.*

On the other hand, the presented methods do not *establish a reduction from the Rabin fragment (monadic first-order sentences with equality and a single unary function symbol, cf. page 23) to the Shelah fragment. The problem is equations that do not adhere to the syntactic restrictions of GAF. It seems that these cannot be treated by the same methods we have employed to deal with equations in the monadic fragment in the proof of Theorem 3.1.5, where we devised an equivalence-preserving translation from $\mathrm{MFO}_{\approx}$ into BSR.*

At this point we have settled the question concerning decidability of GAF-Sat, also under certain syntactic extensions. In fact, decidability of GAF-Sat without any syntactic extensions is already a corollary of the decidability of the satisfiability problem for Maslov's fragment K. The reaon is that the latter syntactically subsumes GAF.

Proposition 3.8.8. *GAF is contained in Maslov's fragment K.*

Proof. Let $\varphi := \forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\bar{\mathrm{u}}_1 \ldots \forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n\bar{\mathrm{u}}_n.\, \psi$ be any GAF sentence with quantifier-free ψ. Recall that φ is relational and does not contain equality. Let the sets At, $\bar{\mathrm{x}}$, $\bar{\mathrm{y}}$, $\bar{\mathrm{u}}$ be defined as in Definition 3.7.1 and let $\mathrm{At}_0, (\mathrm{At}_x)_{x\in\bar{\mathrm{x}}}$ be the partition of the set At described in Definition 3.7.1. Then, in the terminology of the definition of Maslov's fragment K (cf. page 25), the terminal φ-prefix of any atom $A \in \mathrm{At}_x$ with $x \in \bar{\mathrm{x}}$ is a subsequence of $\forall x\exists\bar{\mathrm{u}}_{\mathrm{idx}(x)} \ldots \bar{\mathrm{u}}_n$. Therefore, the terminal φ-prefix of A either ends with an existential quantifier or it is of length one. Regarding the φ-prefix of any atom $B \in \mathrm{At}_0$, we get $\exists\bar{\mathrm{y}}_1 \ldots \bar{\mathrm{y}}_n$. Hence, the terminal φ-prefix of B is empty. Consequently, φ satisfies the conditions of the definition of Maslov's fragment K. □

Of course, Proposition 3.8.8 fails for any extensions of GAF with either equality or non-constant function symbols. We shall see in the next section, how GAF can be extended in such a way that we obtain a generalization of the Gödel–Kalmár–Schütte fragment. Although the latter is syntactically contained in Maslov's fragment K as well, its extension will not (cf. Proposition 3.9.4).

We have not yet given any lower bounds on the blowup that we incur when translating GAF sentences into equivalent Ackermann sentences. However, known bounds regarding the computational complexity of AF-Sat and MFO-Sat give some evidence that this blowup is at least exponential. On the one hand, it is known that the satisfiability problem for AF (without equality) is decidable in deterministic exponential time, even in the presence of arbitrary function symbols (see [BGG97], Theorem 6.3.26 for the former case and Theorem 6.3.1 for the latter). In other words, AF-Sat lies in ExpTime. On the other hand, NExpTime-hardness for MFO-Sat has been shown (cf. Theorem 6.2.13 in [BGG97]). Since MFO is a subfragment of GAF, this entails the following conditional lower bound.

Proposition 3.8.9. *In the worst case, there is at least a super-polynomial blowup in formula length when translating GAF sentences into equivalent AF sentences in a uniform algorithmic way, unless* ExpTime = NExpTime

In Section 4.3, we present a model-theoretic approach including a direct construction of *finite* models for satisfiable GAF sentences. That approach facilitates deriving an upper bound on the size of small models, which in the end also leads to upper bounds on the computational complexity of GAF-Sat.

Remark 3.8.10. *There is also a probabilistic proof for the decidability of the Gödel-Kalmár-Schütte fragment known [GS83], see also Section 6.2.3 in [BGG97]. Since GKS is a syntactic extension of the Ackermann fragment, the proof shows decidability for the latter as well. Although the arguments are indirectly applicable to GAF, via the translation to Ackermann sentences, it might be worthwhile to check whether the probabilistic approach can be applied to GAF sentences directly. We may have to guess some parameters of the probabilistic construction or use upper bounds derivable from what we already know about satisfiable GAF sentences.*

3.9 The Generalized Gödel–Kalmár–Schütte Fragment (GGKS)

It is only a tiny step from the Ackermann fragment to the Gödel–Kalmár–Schütte fragment: simply allow two consecutive universal quantifiers in the quantifier prefix instead of only one. We will see shortly, that, if one views the definition of the generalized Ackermann fragment from the right angle, it is a similarly small step to go from GAF to a generalization of the Gödel–Kalmár–Schütte fragment, which we shall call the *generalized Gödel–Kalmár–Schütte fragment (GGKS)*. Intuitively speaking, a GGKS sentence is of the form $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \bar{\mathrm{u}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n \bar{\mathrm{u}}_n . \psi$ with quantifier-free ψ and that satisfies the following properties. Each atom in φ contains only variables from some subsequence of φ's quantifier prefix of the form $\exists^* \forall \forall \exists^*$. We allow only fixed pairs of universally quantified variables to co-occur in atoms. Any two atoms that are associated with the same pair have the same $\exists^* \forall \forall \exists^*$-subsequence as source of all their variables. The same applies to any two atoms that share some variable from the trailing $\exists^*$-block of their respective quantifier subsequence.

Definition 3.9.1 (Generalized Gödel–Kalmár–Schütte fragment (GGKS))**.**
Let $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \bar{\mathrm{u}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n \bar{\mathrm{u}}_n . \psi$ be a relational first-order sentence without equality. Let At *be the set of all atoms occurring in φ and let $\bar{\mathrm{x}} := \bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_n$, $\bar{\mathrm{y}} := \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_n$, and $\bar{\mathrm{u}} := \bar{\mathrm{u}}_1 \cup \ldots \cup \bar{\mathrm{u}}_n$. Like in Definition 3.7.1, we define the* index of a variable $v \in \bar{\mathrm{x}} \cup \bar{\mathrm{y}} \cup \bar{\mathrm{u}}$ *by* $\mathrm{idx}(v) := k$ *if and only if $v \in \bar{\mathrm{x}}_k \cup \bar{\mathrm{y}}_k \cup \bar{\mathrm{u}}_k$. The sentence φ belongs to the* generalized Gödel–Kalmár–Schütte fragment (GGKS) *if and only if the following conditions are satisfied.* idx(v)

(i) There is some partition of the variables in $\bar{\mathrm{x}}$ into a sequence $\mathcal{X} := \{x_1, x'_1\} \ldots \{x_m, x'_m\}$ of nonempty, pairwise disjoint sets with at most two variables each ($x_i = x'_i$ is allowed). $\mathcal{X}$

(ii) There is a partition of At *into sets* $\mathrm{At}_\emptyset$ *and* $\mathrm{At}_{x,x'}$ *with* $\{x,x'\} \in \mathcal{X}$*, such that the following requirements are met:*

- *(ii.a)* $\mathrm{vars}(\mathrm{At}_\emptyset) \subseteq \bar{\mathrm{y}}$*;*
- *(ii.b) for every* $\{x,x'\} \in \mathcal{X}$ *with* $\mathrm{idx}(x) \leq \mathrm{idx}(x')$ *we have* $\mathrm{vars}(\mathrm{At}_{x,x'}) \subseteq \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_{\mathrm{idx}(x)-1} \cup \{x,x'\} \cup \bar{\mathrm{u}}_{\mathrm{idx}(x')} \cup \ldots \cup \bar{\mathrm{u}}_n$*;*
- *(ii.c) for all distinct sets* $\{x_1,x_1'\},\{x_2,x_2'\} \in \mathcal{X}$ *we have* $\mathrm{vars}(\mathrm{At}_{x_1,x_1'}) \cap \mathrm{vars}(\mathrm{At}_{x_2,x_2'}) \cap \bar{\mathrm{u}} = \emptyset$.

Since the tuples $\bar{\mathrm{x}}_i$ and $\bar{\mathrm{y}}_i, \bar{\mathrm{u}}_i$ in any GGKS sentence φ may be empty, φ's quantifier prefix does not have to start with a universal quantifier and it does not have to end with an existential quantifier. Moreover, notice that every variable $u \in \bar{\mathrm{u}}$ that occurs in φ is associated with exactly one set $\{x,x'\} \in \mathcal{X}$ containing at least one and at most two *reference variables* $x,x' \in \bar{\mathrm{x}}$, determined by the set $\mathrm{At}_{x,x'}$ in which u occurs. Intuitively speaking, like in the case of GAF, any quantifier $\exists u$ with $u \in \bar{\mathrm{u}}$ can be shifted out of the scope of any universal quantifier that does not bind one of u's reference variables.

Deciding membership in GGKS for a given sentence can be done deterministically in polynomial time. The procedure is based on the concepts that we already used for deciding membership in GAF.

Theorem 3.9.2. *Deciding whether a given first-order sentence belongs to GGKS can be done deterministically in time that is polynomial in the length of any reasonable encoding of the input sentence.*

Proof sketch. We only slightly adapt the proof of Theorem 3.7.3. Let $\varphi := \forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{v}}_1 \ldots \forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{v}}_n.\, \psi$ be any relational first-order sentence in prenex normal form with quantifier-free ψ. Let $\bar{\mathrm{x}} := \bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_n$ and $\bar{\mathrm{v}} := \bar{\mathrm{v}}_1 \cup \ldots \cup \bar{\mathrm{v}}_n$. For the moment we do neither know a priori how the variables in each existential quantifier block $\exists\bar{\mathrm{v}}_k$ are to be partitioned into $\bar{\mathrm{y}}_k$ and $\bar{\mathrm{u}}_k$, nor do we know the partition of $\bar{\mathrm{x}}$ into the sequence $\mathcal{X} = \{x_1,x_1'\}\ldots\{x_m,x_m'\}$.

$\mathcal{G}_\varphi$, $\mathcal{C}_v^\uparrow$, $\mathrm{at}(\mathcal{C}_v^\uparrow)$, at_0, at_x, U_x

Let the graph $\mathcal{G}_\varphi := \langle V,E\rangle$ and the *upward closure* $\mathcal{C}_v^\uparrow$ be defined like in the proof of Theorem 3.7.3. In addition, we take over the definitions of $\mathrm{at}(\mathcal{C}_v^\uparrow)$ for $v \in \bar{\mathrm{v}}$ and at_x for $x \in \bar{\mathrm{x}}$ and at_0. Moreover, we reuse the notation $U_x := \mathrm{vars}(\mathrm{at}_x) \cap \bigcup_{i \geq \mathrm{idx}(x)} \bar{\mathrm{v}}_i$.

Claim I: Assume that

(A) for every $x \in \bar{\mathrm{x}}$ there is at most one $x' \in \bar{\mathrm{x}} \setminus \{x\}$ such that $\mathrm{at}_x \cap \mathrm{at}_{x'}$ is nonempty, and

(B) for all distinct $x,x' \in \bar{\mathrm{x}}$ with $\mathrm{idx}(x) \leq \mathrm{idx}(x')$ and nonempty $\mathrm{at}_x \cap \mathrm{at}_{x'}$ we have $v \notin \mathrm{vars}(\mathrm{at}_x \cup \mathrm{at}_{x'})$ for every $v \in \bigcup_{i=\mathrm{idx}(x)}^{\mathrm{idx}(x')-1} \bar{\mathrm{v}}_i$.

$\mathcal{X}'$

Let $\mathcal{X}' := \{x_1,x_1'\},\ldots,\{x_{m'},x_{m'}'\}$ be a maximal sequence of nonempty, pairwise disjoint subsets of $\bar{\mathrm{x}}$, each containing exactly two distinct variables x_i, x_i' for which $\mathrm{at}_{x_i} \cap \mathrm{at}_{x_i'}$ is

$\mathcal{X}$

nonempty. Let $\mathcal{X} := \mathcal{X}'\{x_{m'+1}\}\ldots\{x_m\}$ be a maximal extension of $\mathcal{X}'$ that constitutes a partition of $\bar{\mathrm{x}}$, i.e. every $x \in \bar{\mathrm{x}}$ occur is exactly one set in $\mathcal{X}$. We define the sets $\mathrm{at}_{x,x'} :=$

$\mathrm{at}_{x,x'}$, $U_{x,x'}$, $\mathrm{at}_\emptyset$

$\mathrm{at}_x \cup \mathrm{at}_{x'}$ and $U_{x,x'} := U_x \cup U_{x'}$ for every $\{x,x'\} \in \mathcal{X}$. Moreover, we set $\mathrm{at}_\emptyset := \mathrm{at}_0$. Then, we observer the following properties:

(i) For all distinct $\{x_1,x_1'\},\{x_2,x_2'\} \in \mathcal{X}$ we have $\mathrm{at}_{x_1,x_1'} \cap \mathrm{at}_{x_2,x_2'} = \emptyset$.

(ii) For every $\{x,x'\} \in \mathcal{X}$ we have $\mathrm{vars}(\mathrm{at}_{x,x'}) \cap \bar{\mathrm{x}} = \{x,x'\}$.

(iii) For every $\{x,x'\} \in \mathcal{X}$ we have $U_{x,x'} \cap \mathrm{vars}(\mathrm{at}_\emptyset) = \emptyset$.

(iv) For all distinct $\{x_1,x_1'\},\{x_2,x_2'\} \in \mathcal{X}$ with $\min\big(\mathrm{idx}(x_1),\mathrm{idx}(x_1')\big) \leq \min\big(\mathrm{idx}(x_2),\mathrm{idx}(x_2')\big)$ we have $U_{x_1,x_1'} \cap \mathrm{vars}(\mathrm{at}_{x_2,x_2'}) = \emptyset$.

Proof:

Ad (i): Suppose there are variables x_1, x_2 stemming from distinct sets $\{x_1, x_1'\}, \{x_2, x_2'\} \in \mathcal{X}$ and there is some atom $A \in \mathrm{at}_{x_1} \cap \mathrm{at}_{x_2}$. Then, $x_1 \neq x_2$. Moreover, by maximality of $\mathcal{X}'$, we must have $\{x_1, x_2\}$ as one element in $\mathcal{X}$. Hence, $\{x_1, x_2\}$, $\{x_1, x_1'\}$, and $\{x_2, x_2'\}$ are distinct, and Condition (A) entails $x_1 = x_1'$ and $x_2 = x_2'$. But this contradicts the requirement that all sets in $\mathcal{X}$ are pairwise disjoint.

Ad (ii): This is a direct consequence of (i) and the definition of at_x, $\mathrm{at}_{x'}$, and $\mathrm{at}_{x,x'}$.

Ad (iii): Whenever $v \in U_{x,x'} \subseteq \bigcup_{i \geq \min\left(\mathrm{idx}(x), \mathrm{idx}(x')\right)} \bar{\mathrm{v}}_i$ we have that $\mathrm{at}(\mathcal{C}_v^{\uparrow}) \subseteq \mathrm{at}_{x,x'}$. Suppose there is some $v \in U_{x,x'} \cap \mathrm{vars}(\mathrm{at}_\emptyset)$, i.e. there is some $A \in \mathrm{at}_\emptyset = \mathrm{at}_0$ with $v \in \mathrm{vars}(A)$. Since $A \in \mathrm{at}(\mathcal{C}_v^{\uparrow}) \subseteq \mathrm{at}_{x,x'}$, by definition of at_0, A cannot occur in at_0 and, hence, not in $\mathrm{at}_\emptyset$.

Ad (iv): Whenever $v \in U_{x_1,x_1'} \subseteq \bigcup_{i \geq \min\left(\mathrm{idx}(x_1), \mathrm{idx}(x_1')\right)} \bar{\mathrm{v}}_i$, we observe that $\mathrm{at}(\mathcal{C}_v^{\uparrow}) \subseteq \mathrm{at}_{x_1,x_1'}$. Suppose there is some $v \in U_{x_1,x_1'} \cap \mathrm{vars}(\mathrm{at}_{x_2,x_2'})$. Hence, there must be some atom $A \in \mathrm{at}_{x_2,x_2'}$ in which v occurs. But since A belongs to $\mathrm{at}(\mathcal{C}_v^{\uparrow})$, we know that $A \in \mathrm{at}_{x_1,x_1'}$. This contradicts (i). ◊

Claim II: The sentence φ belongs to GGKS if and only if it satisfies Conditions (A) and (B) from Claim I.

Proof: Regarding the *if*-direction, we construct the sequence $\mathcal{X}$ as described in the proof of Claim I, we set $\mathrm{At}_\emptyset := \mathrm{at}_\emptyset$ and for every $\{x, x'\} \in \mathcal{X}$ we set $\mathrm{At}_{x,x'} := \mathrm{at}_{x,x'}$. Moreover, we partition every existential quantifier block $\exists \bar{\mathrm{v}}_k$ into $\exists \bar{\mathrm{y}}_k \exists \bar{\mathrm{u}}_k$ by setting $\bar{\mathrm{u}}_k := \bar{\mathrm{v}}_k \cap \bigcup_{\{x,x'\} \in \mathcal{X}} U_{x,x'}$ and $\bar{\mathrm{y}}_k := \bar{\mathrm{v}}_k \setminus \bar{\mathrm{u}}_k$. Condition (i) of Definition 3.9.1 is certainly satisfied by $\mathcal{X}$. Condition (ii.a) of Definition 3.9.1 is satisfied due to the following observations. By definition of $\mathrm{at}_\emptyset$ and the $\mathrm{at}_{x,x'}$, we have $\mathrm{vars}(\mathrm{At}_\emptyset) \cap \bar{\mathrm{x}} = \emptyset$. By virtue of Claim I(iii) and the above partition of the $\bar{\mathrm{v}}_k$ into $\bar{\mathrm{y}}_k$ and $\bar{\mathrm{u}}_k$, we have $\bar{\mathrm{u}}_k \cap \mathrm{vars}(\mathrm{At}_\emptyset) = \emptyset$ for every k. Hence, $\mathrm{vars}(\mathrm{At}_\emptyset) \subseteq \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_n$. Condition (ii.b) of Definition 3.9.1 is a consequence of the way we partition the $\bar{\mathrm{v}}_k$ into $\bar{\mathrm{y}}_k, \bar{\mathrm{u}}_k$. By Condition (B), any variable $v \in \bar{\mathrm{v}} \cap \mathrm{vars}(\mathrm{At}_{x,x'})$ with index $\mathrm{idx}(v) = k$ belongs to $U_{x,x'}$ if and only if $k \geq \max\big(\mathrm{idx}(x), \mathrm{idx}(x')\big)$. Hence, again by Condition (B), we have $v \in \bar{\mathrm{y}}_k$ if and only if $k < \min\big(\mathrm{idx}(x), \mathrm{idx}(x')\big)$, and we have $v \in \bar{\mathrm{u}}_k$ otherwise. Moreover, Claim I(ii) states that x, x' are the only variables from $\bar{\mathrm{x}}$ that occur in $\mathrm{At}_{x,x'}$. Condition (ii.c) of Definition 3.9.1 follows immediately from Claim I(iv) and the the fact that $\bar{\mathrm{u}}_1 \cup \ldots \cup \bar{\mathrm{u}}_n = \bigcup_{\{x,x'\} \in \mathcal{X}} U_{x,x'}$.

$\mathrm{At}_{x,x'}$, $\mathrm{At}_\emptyset$

Regarding the *only if*-direction we argue as follows. Condition (A) of Claim I is certainly satisfied, if Conditions (ii.b) and (ii.c) of Definition 3.9.1 are met by φ. Consider any pair of distinct variables $x, x' \in \bar{\mathrm{x}}$ with $\mathrm{idx}(x) \leq \mathrm{idx}(x')$ and nonempty $\mathrm{at}_x \cap \mathrm{at}_{x'}$. By construction of at_x and $\mathrm{at}_{x'}$, a nonempty intersection of the two entails that the set $\mathrm{at}_x \cup \mathrm{at}_{x'}$ cannot be partitioned into two parts $\mathrm{at}_1, \mathrm{at}_2$ such that at_1 contains x, at_2 contains x', and at_1 and at_2 do not share any variables from $\{x, x'\} \cup \bar{\mathrm{u}}_{\mathrm{idx}(x)} \cup \ldots \cup \bar{\mathrm{u}}_n$. Hence, by Conditions (ii.b) and Conditions (ii.c) of Definition 3.9.1, we must have $\{x, x'\} \in \mathcal{X}$. But then, Condition (ii.b) of Definition 3.9.1 entails that $\mathrm{vars}(\mathrm{at}_x \cup \mathrm{at}_{x'}) \cap \bar{\mathrm{u}}_k = \mathrm{vars}(\mathrm{At}_{x,x'}) \cap \bar{\mathrm{u}}_k$ is empty for every k with $\mathrm{idx}(x) \leq k < idx(x')$. This entails that Condition (B) of Claim I is satisfied. ◊

By Claim II, Conditions (A) and (B) from Claim I together yield a criterion to decide whether φ belongs to GGKS or not. It remains to argue that this criterion can be checked deterministically in polynomial time. We have already argued in the proof of Theorem 3.7.3 that the graph $\mathcal{G}_\varphi$ and the sets at_x can be computed deterministically in time that is polynomial in $\|\varphi\|$. The sum of the lengths of the atoms in all the at_x taken together is at most $|\bar{\mathrm{x}}| \cdot \mathrm{len}(\varphi) \leq \big(\mathrm{len}(\varphi)\big)^2$. Therefore, checking whether $\mathrm{at}_x \cap \mathrm{at}_{x'}$ for any x, x' is empty and, hence, checking Conditions (A) and (B) can be done in time polynomial in $\|\varphi\|$. □

GGKS obviously contains sentences that GAF does not, e.g. $\forall x_1 x_2.\, P(x_1, x_2)$. It is also easy to see that GGKS is an extension of GAF: if we restrict the sequences $\mathcal{X}$ so that they contain only

singleton sets, then we essentially obtain the definition of GAF. Hence, also AF and MFO are a subset of GGKS. Finally, consider any GKS sentence $\exists\bar{y}\forall x_1x_2\exists\bar{u}.\,\psi$ with quantifier-free ψ. We define the sequence $\mathcal{X} := \{x_1, x_2\}$ and let At_{x_1,x_2} be the set of all atoms occurring in φ. Then, $\mathcal{X}$ and At_{x_1,x_2} satisfy the conditions of Definition 3.9.1 and thus witness that φ belongs to GGKS.

Proposition 3.9.3. *GGKS properly contains GKS, GAF, AF, and MFO.*

In the previous section we have seen that GAF is contained in Maslov's fragment K. We shall see now that we left the realm of the latter class with the step from GAF to GGKS.

Proposition 3.9.4. *GGKS and Maslov's fragment K are syntactically incomparable classes of sentences.*

Proof. The following sentence witnesses that GGKS is not contained in Maslov's fragment K:

$$\forall x_1x_1'x_2x_2'.\, P(x_1, x_1') \vee Q(x_2, x_2') \ .$$

The sentence belongs to GGKS but not to Maslov's class K. On the other hand, it is easy to find sentences that belong to K but not to GGKS, e.g.

$$\forall x_1x_2x_3\exists y_1y_2.\, P(x_1, x_2, y_1) \wedge Q(x_3, y_1, y_2) \wedge R(x_1, x_2, x_3) \ . \qquad \square$$

Next, we sketch an equivalence-preserving translation from GGKS into GKS. The bulk of the work was already described in Section 3.8. Again, we proceed in two stages, first transforming a given GGKS sentence into *GGKS special form* and, afterwards, into an equivalent GKS sentence.

Lemma 3.9.5. *If φ belongs to GGKS, we can effectively construct an equivalent sentence φ' in standard form, in which every subformula lies within the scope of at most two universal quantifiers, and the scope of every universal quantifier contains at most one more universal quantifier. Moreover, all literals in φ' occur in φ (modulo variable renaming).*

The transformation mentioned in the lemma is essentially a slight adaptation of the analogous transformation for the GAF case (cf. Lemma 3.8.1). The sentence φ' can easily be further transformed into a particular shape to which we shall refer as *GGKS special form*:

GGKS special form

$$\exists\bar{z}.\bigwedge_i\Big(\bigvee_j \forall x_{i,j}x'_{i,j}\exists\bar{y}_{i,j}.\,\chi_{i,j}(\bar{z}, x_{i,j}, x'_{i,j}, \bar{y}_{i,j})\Big) \vee \eta_i(\bar{z})$$

where the $\chi_{i,j}$ and the η_i are quantifier free. One can show that every such sentence is equivalent to some GKS sentence. Therefore, every GGKS sentence is equivalent to a GKS sentence.

Lemma 3.9.6. *Every GGKS sentence φ in GGKS special form can be effectively transformed into an equivalent sentence φ' that has the shape $\exists\bar{z}\,\forall xx'\exists\bar{y}.\,\psi$ with quantifier-free ψ.*

Proof. Since φ is in GGKS special form, it has the shape

$$\varphi'' := \ \exists\bar{z}.\bigwedge_i\big(\bigvee_j \forall x_{i,j}x'_{i,j}\exists\bar{y}_{i,j}.\,\chi_{i,j}(\bar{z}, x_{i,j}, x'_{i,j}, \bar{y}_{i,j})\big) \vee \eta_i(\bar{z}),$$

where the $\chi_{i,j}$ and the η_i are quantifier free. Consider any subformula of the form $\psi' := \bigvee_j \forall xx'\,\exists\bar{y}.\,\chi_j(\bar{z}, x, x', \bar{y})$, possibly containing free variables from $\bar{z}$. By virtue of Lemma 3.8.3, ψ' is equivalent to some formula of the form $\exists\bar{v}'\bar{y}'.\,\chi'(\bar{z}, \bar{v}', \bar{y}') \wedge \forall xx'\exists\bar{y}.\,\chi''(\bar{z}, x, x', \bar{y}, \bar{v}', \bar{y}')$ with quantifier-free χ', χ''. Hence, φ'' is equivalent to some sentence that, after shifting some quantifiers outwards, is of the form $\exists\bar{z}.\bigwedge_i\big(\exists\bar{u}_i\forall x_ix'_i\exists\bar{w}_i.\,\psi''_i(\bar{z}, \bar{u}_i, x_i, x'_i, \bar{w}_i)\big) \vee \eta_i(\bar{z})$, where the ψ''_i and the η_i are quantifier free. A prenex version of this sentence yields the sought φ'. $\square$

Theorem 3.9.7. *Every GGKS sentence is equivalent to some GKS sentence.*

Since we know that GKS enjoys the finite model property and, hence, the decidability problem for GKS is decidable, this result immediately entails decidability of the satisfiability problem for GGKS (*GGKS-Sat*).

GGKS-Sat

Corollary 3.9.8. *The satisfiability problem for GGKS is decidable, and GGKS enjoys the finite model property.*

As we have already pointed out in Remark 3.8.10, Gurevich and Shelah [GS83] gave a probabilistic proof for the decidability of GKS, see also Section 6.2.3 in [BGG97]. It would be interesting to approach decidability of GGKS-Sat using a probabilistic approach without relying on the translation from GGKS to GKS.

We finish the present section emphasizing that GGKS sentences can be substantially more succinct than equivalent GKS sentences. The following theorem formulates a lower bound regarding the incurred blowup that comes along with any equivalence-preserving translation from GGKS to GKS.

Theorem 3.9.9. *There is a class of GGKS sentences and some positive integer n_0 such that for every integer $n \geq n_0$ the class contains a sentence φ with a length linear in n for which any equivalent GKS sentence has a length that is at least exponential in n.*

Proof sketch. Let $n \geq 1$ be some positive integer. Consider the following first-order sentence in which the sets $\{x_1, x_2\}$ and $\{y_1, y_2\}$ are separated:

$$\varphi := \forall x_2 \exists y_2 \forall x_1 \exists y_1 . \bigwedge_{i=1}^{8n} \big(P_i(x_1, x_2) \leftrightarrow Q_i(y_1, y_2)\big) \ .$$

In analogy to the proof of Theorem 3.2.7, we construct the following model $\mathcal{A}$ for φ. The construction is based on the sets $\mathcal{S}_1 := \big\{S \subseteq [8n] \;\big|\; |S| = 2n\big\}$ and $\mathcal{S}_2 := \big\{S \subseteq \mathcal{S}_1 \;\big|\; |S| = \frac{1}{2}|\mathcal{S}_1|\big\}$. $\mathcal{S}_k$
We observe that

$$|\mathcal{S}_1| = \binom{8n}{2n} \;\geq\; \left(\frac{8n}{2n}\right)^{2n} \;=\; 2^{4n} \qquad \text{and} \qquad |\mathcal{S}_2| = \binom{|\mathcal{S}_1|}{|\mathcal{S}_1|/2} \;\geq\; \left(\frac{|\mathcal{S}_1|}{|\mathcal{S}_1|/2}\right)^{|\mathcal{S}_1|/2} \;\geq\; 2^{2^{4n-1}} \ .$$

Having the sets $\mathcal{S}_1, \mathcal{S}_2$, we now define the structure $\mathcal{A}$ as follows: $\mathcal{A}$

$$\mathsf{A} := \big\{\mathsf{a}_S^{(1)}, \mathsf{b}_S^{(1)} \;\big|\; S \in \mathcal{S}_1\big\} \cup \big\{\mathsf{a}_S^{(2)}, \mathsf{b}_S^{(2)} \;\big|\; S \in \mathcal{S}_2\big\},$$

$$P_i^{\mathcal{A}} := \big\{\langle \mathsf{a}_{S_1}^{(1)}, \mathsf{a}_{S_2}^{(2)}\rangle \in \mathsf{A}^2 \;\big|\; i \in S_1 \in S_2\big\} \text{ for } i = 1, \ldots, 8n, \text{ and}$$

$$Q_i^{\mathcal{A}} := \big\{\langle \mathsf{b}_{S_1}^{(1)}, \mathsf{b}_{S_2}^{(2)}\rangle \in \mathsf{A}^n \;\big|\; i \in S_1 \in S_2\big\} \text{ for } i = 1, \ldots, 8n.$$

Clearly, for any choice of S_1, S_2 and every i, $1 \leq i \leq 8n$, we have

$$\mathcal{A}, \big[x_1 \mapsto \mathsf{a}_{S_1}^{(1)}, x_2 \mapsto \mathsf{a}_{S_2}^{(2)}, y_1 \mapsto \mathsf{b}_{S_1}^{(1)}, y_2 \mapsto \mathsf{b}_{S_2}^{(2)}\big] \models P_i(x_1, x_2) \leftrightarrow Q_i(y_1, y_2) \ .$$

For any other choice of pairs $\langle \mathsf{c}_1, \mathsf{c}_2\rangle$, i.e. there do not exist sets $S_1 \in \mathcal{S}_1$ and $S_2 \in \mathcal{S}_2$ such that $\langle \mathsf{c}_1, \mathsf{c}_2\rangle$ equals $\langle \mathsf{a}_{S_1}^{(1)}, \mathsf{a}_{S_2}^{(2)}\rangle$ or $\langle \mathsf{b}_{S_1}^{(1)}, \mathsf{b}_{S_2}^{(2)}\rangle$, we observe $\mathcal{A}, [x_1 \mapsto \mathsf{c}_1, x_2 \mapsto \mathsf{c}_2] \not\models P_i(x_1, x_2)$ and $\mathcal{A}, [y_1 \mapsto \mathsf{c}_1, y_2 \mapsto \mathsf{c}_2] \not\models Q_i(y_1, y_2)$ for every i. Hence,

$$\mathcal{A}, [x_1 \mapsto \mathsf{c}_1, x_2 \mapsto \mathsf{c}_2, y_1 \mapsto \mathsf{c}_1, y_2 \mapsto \mathsf{c}_2] \models \bigwedge_{i=1}^{8n} P_i(x_1, x_2) \leftrightarrow Q_i(y_1, y_2) \ .$$

Consequently, $\mathcal{A}$ is a model of φ.

For every $S \in \mathcal{S}_1 \cup \mathcal{S}_2$ we define the structure $\mathcal{A}_{-S}$ as the substructure of $\mathcal{A}$ induced by the $\mathcal{A}_{-S}$
domain $\mathsf{A}_{-S} := \mathsf{A} \setminus \{\mathsf{b}_S^{(k)}\}$, where $k = 1$ if $S \in \mathcal{S}_1$ and $k = 2$ if $S \in \mathcal{S}_2$. Like in the proof of Theorem 3.2.7 we can prove the following claim.

Claim I: For every $S \in \mathcal{S}_1 \cup \mathcal{S}_2$ the substructure $\mathcal{A}_{-S}$ of $\mathcal{A}$ does not satisfy φ. $\Diamond$

Let $\varphi_* := \exists\bar{z}\forall x_1x_2\exists\bar{y}.\,\psi_*$ with quantifier-free ψ_* be a shortest GKS sentence equivalent to φ. Suppose that the length of φ_* is less than 2^n. Let $\psi' := \bigvee_{i\in I}\chi_i(\bar{z},x_1,x_2,\bar{y})$ be a shortest disjunction of conjunctions of literals that is equivalent to ψ_*. We observe that the index set I contains fewer than 2^{2^n} indices, for otherwise we could find a shorter formula satisfying our requirements. For the same reason every conjunction χ_i contains at most 2^n literals.

Let $\bar{\mathsf{d}}$ be some tuple for which we have

$$\mathcal{A},[\bar{z}\mapsto\bar{\mathsf{d}}]\models\forall x_1x_2\exists\bar{y}.\bigvee_{i\in I}\chi_i(\bar{z},x_1,x_2,\bar{y})\ .$$

D Let $\mathsf{D} := \{\mathsf{b}_S^{(2)} \mid S\in\mathcal{S}_2 \text{ and } \mathsf{b}_S^{(2)}\notin\bar{\mathsf{d}}\}$. Because of $|\bar{\mathsf{d}}|\le|\bar{z}|\le\mathrm{len}(\varphi_*)\le 2^n$ and $|\mathcal{S}_2|\ge 2^{2^{4n-1}}\ge 2^{2^{3n}}$, we have $|\mathsf{D}|\ge 2^{2^{2n}}$ for sufficiently large n. By Claim I, we observe

$$\mathcal{A}_{-S},[\bar{z}\mapsto\bar{\mathsf{d}}]\not\models\forall x_1x_2\exists\bar{y}.\bigvee_{i\in I}\chi_i(\bar{z},x_1,x_2,\bar{y})$$

for every S with $\mathsf{b}_S^{(2)}\in\mathsf{D}$. Hence, for every $\mathsf{b}_S^{(2)}\in\mathsf{D}$ there is some pair $\mathsf{c}_1,\mathsf{c}_2\in\mathsf{A}\setminus\{\mathsf{b}_S^{(2)}\}$, some tuple $\bar{\mathsf{b}}$ containing $\mathsf{b}_S^{(2)}$ and some index $i_S\in I$ such that

$$\mathcal{A},[\bar{z}\mapsto\bar{\mathsf{d}},x_1\mapsto\mathsf{c}_1,x_2\mapsto\mathsf{c}_2,\bar{y}\mapsto\bar{\mathsf{b}}]\models\chi_i(\bar{z},x_1,x_2,\bar{y})$$

and

$$\mathcal{A}_{-S},[\bar{z}\mapsto\bar{\mathsf{d}},x_1\mapsto\mathsf{c}_1,x_2\mapsto\mathsf{c}_2]\not\models\exists\bar{y}.\,\chi_i(\bar{z},x_1,x_2,\bar{y})\ .$$

Because of $|I|\le 2^{2^n}$ and $|\mathsf{D}|\ge 2^{2^{2n}}$, there must be some index i_* that appears as i_S for at least

$$\frac{|\mathsf{D}|}{|I|}\ge\frac{2^{2^{2n}}}{2^{2^n}}=2^{2^{2n}-2^n}\ge 2^{(2^n)^2/2^n}=2^{2^n}$$

D_* elements $\mathsf{b}_S^{(2)}\in\mathsf{D}$, in case n is sufficiently large. Let $\mathsf{D}_*\subseteq\mathsf{D}$ be the set that comprises exactly those elements. In other words, we have $|\mathsf{D}_*|\ge 2^{2^n}$ and for every $\mathsf{b}_S^{(2)}\in\mathsf{D}_*$ there is some pair $\mathsf{c}_1,\mathsf{c}_2$ and some tuple $\bar{\mathsf{b}}$ containing $\mathsf{b}_S^{(2)}$ such that

$$\mathcal{A},[\bar{z}\mapsto\bar{\mathsf{d}},x_1\mapsto\mathsf{c}_1,x_2\mapsto\mathsf{c}_2,\bar{y}\mapsto\bar{\mathsf{b}}]\models\chi_{i_*}(\bar{z},x_1,x_2,\bar{y}) \tag{3.7}$$

and

$$\mathcal{A}_{-S},[\bar{z}\mapsto\bar{\mathsf{d}},x_1\mapsto\mathsf{c}_1,x_2\mapsto\mathsf{c}_2]\not\models\exists\bar{y}.\,\chi_{i_*}(\bar{z},x_1,x_2,\bar{y})\ . \tag{3.8}$$

Consider some $\mathsf{b}_S^{(2)}\in\mathsf{D}_*$ with $S\in\mathcal{S}_2$ and fix it. The only atoms in χ_{i_*} that could possibly contribute to the effect described in (3.7) and (3.8) for $\mathsf{b}_S^{(2)}$ have the form $Q_j(z,y')$, $Q_j(y,y')$, $Q_j(x_1,y')$, or $Q_j(x_2,y')$ for $z\in\bar{z}$, $y,y'\in\bar{y}$, and $1\le j\le 8n$, and, moreover, the variables z,y,x_1,x_2 need to be assigned values $\mathsf{b}_T^{(1)}$ with $T\in\mathcal{S}_1$. Let $\mathcal{S}_1'$ be the set collecting all the T from $\mathcal{S}_1$ that are assigned to such variables occurring in atoms of the mentioned kind. As χ_{i_*} contains at most 2^n such variables, $|\mathcal{S}_1'|\le 2^n$. Recall that S contains $\frac{1}{2}|\mathcal{S}_1|\ge 2^{4n-1}$ sets of indices. By construction of $\mathcal{S}_2$, there must be some some $S'\in\mathcal{S}_2$ such that for every $T\in\mathcal{S}_1'$ we have $T\in S'$ if and only if $T\in S$. Let $\bar{\mathsf{b}}'$ be the tuple that results from $\bar{\mathsf{b}}$ by replacing every occurrence of $\mathsf{b}_S^{(2)}$ in the tuple by $\mathsf{b}_{S'}^{(2)}$. Then, we get $\mathcal{A}_{-S},[\bar{z}\mapsto\bar{\mathsf{d}},x_1\mapsto\mathsf{c}_1,x_2\mapsto\mathsf{c}_2,\bar{y}\mapsto\bar{\mathsf{b}}']\models\chi_{i_*}(\bar{z},x_1,x_2,\bar{y})$, which contradicts (3.8). Consequently, the length of the sentence φ_* cannot be less than 2^n. □

3.10 Separateness and Guarded Quantification

In the beginning of Chapter 3 — more precisely, on page 26 —, we have briefly introduced the concept of guarded quantification. The idea is that a quantifier $\mathcal{Q}\bar{u}$ is not only accompanied by its

scope $\psi(\bar{u}, \bar{v})$ but also by a *guard* $\gamma(\bar{u}, \bar{v})$. A guard is a formula that contains all variables that occur freely in the scope $\psi(\bar{u}, \bar{v})$ and satisfies additional syntactic restrictions. For instance, guards may be restricted to atomic formulas — which is characteristic for the *guarded fragment* —, and we then speak of *atomic guards*. Or guards $\gamma(\bar{u}, \bar{v})$ may be restricted to nonempty conjunctions of atoms $A_1(\bar{u}, \bar{v}) \wedge \ldots \wedge A_k(\bar{u}, \bar{v})$ such that every variable $u \in \bar{v}$ co-occurs with every $v \in \bar{u} \cup \bar{v}$ in at least one A_j. This kind of guards of called *loose guards*.

Given any such guard $\gamma(\bar{u}, \bar{v})$, *guarded quantification* has two possible shapes: $\forall \bar{u}.\, \gamma(\bar{u}, \bar{v}) \rightarrow \psi(\bar{u}, \bar{v})$ and $\exists \bar{u}.\, \gamma(\bar{u}, \bar{v}) \wedge \psi(\bar{u}, \bar{v})$, which are dual to one another. If we restrict our attention to first-order sentences in which all quantifiers are guarded in the described way by atomic guards or loose guards, then we are in the realm of the *guarded fragment* or the *loosely guarded fragment*.

Definition 3.10.1 (Guarded fragment (GF) and loosely guarded fragment (LGF))**.** *An* atomic guard $\gamma(\bar{u}, \bar{v})$ *is an atom* $A(\bar{u}, \bar{v})$ *such that all* $u \in \bar{u} \cup \bar{v}$ *occur in* $A(\bar{u}, \bar{v})$*. A* loose guard $\gamma(\bar{u}, \bar{v})$ *is a nonempty conjunction of atoms* $\gamma(\bar{u}, \bar{v}) := A_1(\bar{u}, \bar{v}) \wedge \ldots \wedge A_k(\bar{u}, \bar{v})$ *such that* $\bar{u}$ *is nonempty,* $\bar{u}$ *and* $\bar{v}$ *are disjoint, and all* u, v *with* $u \in \bar{u}$ *and* $v \in \bar{u} \cup \bar{v}$ *co-occur in at least one* A_j*.*

We define the set of loosely guarded formulas *inductively:*

(i) every relational atom is a loosely guarded formula, equality is admitted;

(ii) every Boolean combination of loosely guarded formulas is a loosely guarded formula;

(iii) for all tuples $\bar{u}, \bar{v}$ *and any loose guard* $\gamma(\bar{u}, \bar{v})$ *the following formulas are loosely guarded formulas:*
$\forall \bar{u}.\, \big(\gamma(\bar{u}, \bar{v}) \rightarrow \psi(\bar{u}, \bar{v})\big)$ *— abbreviated by* $\big(\forall \bar{u}.\, \gamma(\bar{u}, \bar{v})\big)\psi(\bar{u}, \bar{v})$ *— and*
$\exists \bar{u}.\, \big(\gamma(\bar{u}, \bar{v}) \wedge \psi(\bar{u}, \bar{v})\big)$ *— abbreviated by* $\big(\exists \bar{u}.\, \gamma(\bar{u}, \bar{v})\big)\psi(\bar{u}, \bar{v})$*.*

Notice that we assume in any loosely guarded formula $\big(\mathcal{Q}\bar{u}.\, \gamma(\bar{u}, \bar{v})\big)\psi(\bar{u}, \bar{v})$ *that (a) all variables that occur freely in* ψ *also occur in* γ *and (b) every variable that is bound by* $\mathcal{Q}\bar{u}$ *co-occurs with every free variable from* ψ *in some atom in* γ*.*

The loosely guarded fragment (LGF) *is the class of all loosely guarded sentences. The* guarded fragment (GF) *is defined in the same way, except that we require all guards to be atomic.*

We shall occasionally use sloppy language and speak of LGF *formulas* when we mean loosely guarded formulas that are not necessarily closed. Formally, LGF exclusively contains sentences. The same applies to GF *formulas*.

At first glance it seems that guarded quantification and separateness of quantified variables are two opposite properties. In particular, any guard $\gamma(\bar{u}, \bar{v})$ in a formula $\big(\forall \bar{u}.\, \gamma(\bar{u}, \bar{v})\big)\varphi(\bar{u}, \bar{v})$ has to ensure that every $u \in \bar{u}$ co-occurs with each $v \in \bar{v}$ in at least one atom in γ. Clearly, this destroys any separateness of variables from $\bar{u}$ and $\bar{v}$ which might be separated in φ. However, it turns out that guardedness and separateness can be conceived as complementing concepts. Combining the two in a certain way can help extending the (loosely) guarded fragment of first-order logic in a way that preserves decidability of the satisfiability problem.

Definition 3.10.2 (Separated loosely guarded fragment (SLGF))**.** *Two tuples* $\bar{x}, \bar{y}$ *are* guard-separated *in a loosely guarded formula* ψ *if for every atom* A *in* ψ *we either have* $\mathrm{vars}(A) \cap \bar{x} = \emptyset$ *or* $\mathrm{vars}(A) \cap \bar{y} = \emptyset$*; the same must hold for every guard* γ *in* ψ*: either* $\mathrm{vars}(\gamma) \cap \bar{x} = \emptyset$ *or* $\mathrm{vars}(\gamma) \cap \bar{y} = \emptyset$*.*

We define the set of separated loosely guarded formulas *inductively as follows. (i) and (ii) are the same as for loosely guarded formulas (cf. Definition 3.10.1). Let* $\bar{u}, \bar{v}, \bar{z}$ *be tuples of variables and let* $\gamma(\bar{u}, \bar{v})$ *be any loose guard.*

(iii) The following are separated loosely guarded formulas: $\forall \bar{u}.\, \big(\gamma(\bar{u}, \bar{v}) \rightarrow \psi(\bar{u}, \bar{v}, \bar{z})\big)$ *and* $\exists \bar{u}.\, \big(\gamma(\bar{u}, \bar{v}) \wedge \psi(\bar{u}, \bar{v}, \bar{z})\big)$*, where the sets* $\bar{u}$ *and* $\bar{z}$ *are guard-separated in* ψ*.*

The separated loosely guarded fragment (SLGF) *is the class of all separated loosely guarded sentences. When we start from the set of* guarded formulas *instead of loosely guarded formulas, we obtain the* separated guarded fragment (SGF)*.*

As for GF and LGF, we shall occasionally use sloppy language and speak of SLGF *formulas* and SGF *formulas* when we mean separated (loosely) guarded formulas that are not necessarily closed.

Remark 3.10.3. *Notice that any formula of the form* $\forall\bar{u}\bar{x}.\,\gamma(\bar{u},\bar{v}) \wedge \delta(\bar{x},\bar{y}) \rightarrow \psi(\bar{u},\bar{v},\bar{x},\bar{y},\bar{z})$ *where* $\bar{u}$, $\bar{x}$, *and* $\bar{z}$ *are pairwise distinct and guard-separated in* ψ *is equivalent to the SLGF formula* $\big(\forall\bar{u}.\,\gamma(\bar{u},\bar{v})\big)\big(\big(\forall\bar{x}.\,\delta(\bar{x},\bar{y})\big)\psi(\bar{u},\bar{v},\bar{x},\bar{y},\bar{z})\big)$. *A dual observation can be made for existential quantification. This means that, under certain restrictions, we can mix variables that are subject to distinct guards in a single quantifier block. One could incorporate this idea into the definition of SLGF and, hence, obtain a syntactically slightly extended version. However, for the sake of simplicity, we adhere to the simpler definition given above.*

It is easy to see that SLGF is indeed a proper syntactic extension of LGF, and that the same applies to SGF and GF. A simple sentence witnessing the strictness of these containment relations is the sentence $\big(\forall x.\,x \approx x\big)\big(\exists y.\,y \approx y\big)\big(P(y) \leftrightarrow \neg P(x)\big)$. It belongs neither to GF nor to LGF, but to both SGF and SLGF. Moreover, the sentence is a witness of the following observation: Every MFO sentence can be easily turned into an equivalent SGF sentence with a length linear in the original. We just need to add trivial equations $v \approx v$ as guards to subformulas $\mathcal{Q}v.\,\chi$. The result of this transformation lies in the intersection of SGF and $\mathrm{MFO}_{\approx}$.

Proposition 3.10.4. *SGF properly contains GF and SLGF properly contains SGF, LGF, and GF. Moreover, every MFO sentence can be turned into an equivalent SGF sentence with a length linear in the original.*

For $\mathrm{MFO}_{\approx}$ sentences the matter seems to be more complicated. The sentence $\forall xy.\,x \approx y$, for instance, is not an SLGF sentence and cannot be directly transformed into an equivalent SLGF sentence in the described way.

Analogously to all the other novel first-order fragments we have defined, there exists an effective translation procedure that transforms SLGF sentences into equivalent LGF sentences.

Lemma 3.10.5. *Every SLGF formula is equivalent to some LGF formula.*

Proof. We prove an auxiliary result from which the lemma follows easily: Consider any SLGF formula $\varphi := \big(\mathcal{Q}\bar{u}.\,\gamma(\bar{u},\bar{v})\big)\psi(\bar{u},\bar{v},\bar{z})$ where ψ is any LGF formula, $\bar{u},\bar{v},\bar{z}$ are pairwise disjoint, and φ's free variables are exactly the ones in $\bar{v},\bar{z}$. Then, φ is equivalent to some LGF formula $\varphi'(\bar{v},\bar{z})$. Moreover, any two sets of variables that are guard-separated in φ are also guard-separated in φ'.

Suppose $\mathcal{Q}$ is a universal quantifier (the case for existential quantification is dual). Recall that, by definition of SLGF, the tuples $\bar{u}$ and $\bar{z}$ need to be guard-separated in ψ. Since ψ is an LGF formula and since we assume that no variable occurs freely and bound in φ at the same time, we know that in every subformula $\chi := \big(\mathcal{Q}\bar{x}.\,\delta(\bar{x},\bar{y})\big)\eta(\bar{x},\bar{y})$ of ψ we either have $\mathrm{vars}(\chi) \cap \bar{u} = \emptyset$ or $\mathrm{vars}(\chi) \cap \bar{z} = \emptyset$ (or both). Moreover, since φ is an SLGF formula, we have $\mathrm{vars}(A) \cap \bar{u} = \emptyset$ or $\mathrm{vars}(A) \cap \bar{z} = \emptyset$ for every atom in ψ. Hence, φ is equivalent to some formula of the form $\varphi'' := \forall\bar{u}.\,\gamma(\bar{u},\bar{v}) \rightarrow \bigwedge_i\big(\chi_i(\bar{u},\bar{v}) \vee \eta_i(\bar{v},\bar{z})\big)$, where the χ_i and η_i are disjunctions of atoms, negated atoms, or LGF formulas of the form $\big(\mathcal{Q}\bar{x}.\,\delta(\bar{x},\bar{y})\big)\eta(\bar{x},\bar{y})$. Applying distributivity and shifting the quantifier $\forall\bar{u}$ in φ'', it is easy to show equivalence to $\varphi' := \bigwedge_i\big(\big(\forall\bar{u}.\,\gamma(\bar{u},\bar{v}) \rightarrow \chi_i(\bar{u},\bar{v})\big) \vee \eta_i(\bar{v},\bar{z})\big)$. This is the sought LGF formula. □

Notice that the proof works irrespectively of the structure of guards. Hence, we also observe that every SGF formula is equivalent to some GF formula.

In connection with the well-known fact that GF and LGF possess the finite model property [Grä99b, Hod02], the obvious consequence of Lemma 3.10.5 is that the satisfiability problems associated with SGF and SLGF (*SGF-Sat* and *SLGF-Sat*) are decidable.

SGF-Sat and *SLGF-Sat*

Theorem 3.10.6. *Both SGF and SLGF possess the finite model property. Moreover, the satisfiability problem for SGF sentences and SLGF sentences is decidable.*

The following example illustrates the translation of SLGF sentences into LGF sentences, using the more liberal SLGF syntax outlined in Remark 3.10.3.

Example 3.10.7. *Let $\bar{u}, \bar{v}, \bar{x}, \bar{y}, \bar{z}$ be five pairwise disjoint tuples of variables and let $\gamma(\bar{u}, \bar{z})$, $\gamma'(\bar{v}, \bar{z})$, $\delta(\overline{xu}, \bar{z})$, $\delta'(\overline{yv}, \bar{z})$ be loose guards, where $\overline{xu}$ and $\overline{yv}$ denote the results of appending $\bar{u}$ to $\bar{x}$ and $\bar{v}$ to $\bar{y}$, respectively. Consider the formula*

$$\exists \bar{u}\bar{v}.\, \gamma(\bar{u}, \bar{z}) \wedge \gamma'(\bar{v}, \bar{z}) \wedge \big(\forall \bar{x}\bar{y}.\, \delta(\overline{xu}, \bar{z}) \wedge \delta'(\overline{yv}, \bar{z}) \rightarrow \varphi(\bar{u}, \bar{v}, \bar{x}, \bar{y}, \bar{z})\big)$$

where φ is quantifier free and the sets $\bar{u} \cup \bar{x}$ and $\bar{v} \cup \bar{y}$ are separated in φ. Clearly, this formula is not loosely guarded, as there are no guarding atoms in which the variables from $\bar{u}$ and $\bar{v}$ co-occur, and the same holds for the variables from $\bar{x}$ and $\bar{y}$. Nevertheless, the formula is equivalent to some LGF formula, as witnessed by the following transformations, where the ψ_i and χ_i are certain disjunctions of literals and the ψ'_j and χ'_j are certain conjunctions of literals and basic formulas:

$$\begin{aligned}
&\exists \bar{u}\bar{v}.\, \gamma(\bar{u}, \bar{z}) \wedge \gamma'(\bar{v}, \bar{z}) \wedge \big(\forall \bar{x}\bar{y}.\, \delta(\overline{xu}, \bar{z}) \wedge \delta'(\overline{yv}, \bar{z}) \rightarrow \varphi(\bar{u}, \bar{v}, \bar{x}, \bar{y}, \bar{z})\big) \\
&\quad \eqqcolon \exists \bar{u}\bar{v}.\, \gamma(\bar{u}, \bar{z}) \wedge \gamma'(\bar{v}, \bar{z}) \wedge \big(\forall \bar{x}\bar{y}.\, \delta(\overline{xu}, \bar{z}) \wedge \delta'(\overline{yv}, \bar{z}) \rightarrow \bigwedge_i \big(\psi_i(\bar{x}, \bar{u}, \bar{z}) \vee \chi_i(\bar{y}, \bar{v}, \bar{z})\big)\big) \\
&\quad \eqqcolon \exists \bar{u}\bar{v}.\, \gamma(\bar{u}, \bar{z}) \wedge \gamma'(\bar{v}, \bar{z}) \wedge \bigwedge_i \big(\forall \bar{x}\bar{y}.\, \delta(\overline{xu}, \bar{z}) \wedge \delta'(\overline{yv}, \bar{z}) \rightarrow \psi_i(\bar{x}, \bar{u}, \bar{z}) \vee \chi_i(\bar{y}, \bar{v}, \bar{z})\big) \\
&\quad \eqqcolon \exists \bar{u}\bar{v}.\, \gamma(\bar{u}, \bar{z}) \wedge \gamma'(\bar{v}, \bar{z}) \wedge \bigwedge_i \Big(\forall \bar{x}\bar{y}.\, \big(\delta(\overline{xu}, \bar{z}) \rightarrow \psi_i(\bar{x}, \bar{u}, \bar{z})\big) \vee \big(\delta'(\overline{yv}, \bar{z}) \rightarrow \chi_i(\bar{y}, \bar{v}, \bar{z})\big)\Big) \\
&\quad \eqqcolon \exists \bar{u}\bar{v}.\, \gamma(\bar{u}, \bar{z}) \wedge \gamma'(\bar{v}, \bar{z}) \wedge \bigwedge_i \Big(\underbrace{\big(\forall \bar{x}.\, \delta(\overline{xu}, \bar{z}) \rightarrow \psi_i(\bar{x}, \bar{u}, \bar{z})\big)}_{\substack{\text{basic formula with free} \\ \text{variables from } \bar{u}, \bar{z}}} \vee \underbrace{\big(\forall \bar{y}.\, \delta'(\overline{yv}, \bar{z}) \rightarrow \chi_i(\bar{y}, \bar{v}, \bar{z})\big)}_{\substack{\text{basic formula with free} \\ \text{variables from } \bar{v}, \bar{z}}}\Big) \\
&\quad \eqqcolon \exists \bar{u}\bar{v}.\, \gamma(\bar{u}, \bar{z}) \wedge \gamma'(\bar{v}, \bar{z}) \wedge \bigvee_j \big(\psi'_j(\bar{u}, \bar{z}) \wedge \chi'_j(\bar{v}, \bar{z})\big) \\
&\quad \eqqcolon \bigvee_j \Big(\exists \bar{u}\bar{v}.\, \gamma(\bar{u}, \bar{z}) \wedge \gamma'(\bar{v}, \bar{z}) \wedge \psi'_j(\bar{u}, \bar{z}) \wedge \chi'_j(\bar{v}, \bar{z})\Big) \\
&\quad \eqqcolon \bigvee_j \Big(\big(\exists \bar{u}.\, \gamma(\bar{u}, \bar{z}) \wedge \psi'_j(\bar{u}, \bar{z})\big) \wedge \big(\exists \bar{v}.\, \gamma'(\bar{v}, \bar{z}) \wedge \chi'_j(\bar{v}, \bar{z})\big)\Big)\ .
\end{aligned}$$

The final result belongs to LGF.

We conclude this section with an investigation of the succinctness gap between SLGF and LGF. The following theorem entails that there is no elementary upper bound on the length of the LGF sentences that result from any equivalence-preserving transformation of SLGF sentences into LGF.

Theorem 3.10.8. *There is a class of SLGF sentences such that for every integer $n \geq 3$ the class contains a sentence φ with n $\forall\exists$ alternations and with a length polynomial in n for which any equivalent LGF sentence has at least $(n-1)$-fold exponential length in n.*

Proof sketch. Let $n \geq 3$. Consider the following SLGF sentence in which the sets $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_n\}$ are separated:

$$\begin{aligned}
\varphi := \big(\forall x_n.\, R_n(x_n)\big)\big(\exists y_n.\, T_n(y_n)\big) \ldots \\
\big(\forall x_1.\, R_1(x_1, \ldots, x_n)\big)\big(\exists y_1.\, T_1(y_1, \ldots, y_n)\big).\, \bigwedge_{i=1}^{4n} \big(P_i(x_1, \ldots, x_n) \leftrightarrow Q_i(y_1, \ldots, y_n)\big)\ .
\end{aligned}$$

In order to construct a particular model of φ, we inductively define the following sets: $\mathcal{S}_1 := \big\{S \subseteq [4n] \;\big|\; |S| = 2n\big\}$, $\mathcal{S}_{k+1} := \big\{S \in \mathcal{P}\mathcal{S}_k \;\big|\; |S| = \frac{1}{2} \cdot |\mathcal{S}_k|\big\}$ for every $k \geq 1$. Hence, we observe that

$$\begin{aligned}
|\mathcal{S}_1| &= \tbinom{4n}{2n} \geq \big(\tfrac{4n}{2n}\big)^{2n} = 2^{2n}, \\
|\mathcal{S}_2| &= \tbinom{|\mathcal{S}_1|}{|\mathcal{S}_1|/2} \geq \big(\tfrac{|\mathcal{S}_1|}{|\mathcal{S}_1|/2}\big)^{|\mathcal{S}_1|/2} = 2^{|\mathcal{S}_1|/2} \geq 2^{2^{2n}/2} = 2^{2^{2n-1}},
\end{aligned}$$

$$\vdots$$

$$|\mathcal{S}_n| \;=\; \tbinom{|\mathcal{S}_{n-1}|}{|\mathcal{S}_{n-1}|/2} \;\geq\; 2^{|\mathcal{S}_{n-1}|/2} \;\geq\; 2^{2^{\cdot^{\cdot^{\cdot^{2^{2n-1}-1}}}}-1} \;\geq\; 2^{\uparrow n}(2n-(n-1)) \;=\; 2^{\uparrow n}(n+1),$$

where the inequality $\binom{n}{k} \geq (n/k)^k$ can be found in [CSRL01] (page 1097), for example.

$\mathcal{A}$

Having the sets $\mathcal{S}_k$, we now define the structure $\mathcal{A}$ as follows:

$$\mathsf{A} := \textstyle\bigcup_{k=1}^{n} \big\{ \mathsf{a}_S^{(k)}, \mathsf{b}_S^{(k)} \;\big|\; S \in \mathcal{S}_k \big\},$$

$$P_i^{\mathcal{A}} := \big\{ \langle \mathsf{a}_{S_1}^{(1)}, \ldots, \mathsf{a}_{S_n}^{(n)} \rangle \in \mathsf{A}^n \;\big|\; i \in S_1 \in S_2 \in \ldots \in S_n \big\} \text{ for } i = 1, \ldots, 4n,$$

$$Q_i^{\mathcal{A}} := \big\{ \langle \mathsf{b}_{S_1}^{(1)}, \ldots, \mathsf{b}_{S_n}^{(n)} \rangle \in \mathsf{A}^n \;\big|\; i \in S_1 \in S_2 \in \ldots \in S_n \big\} \text{ for } i = 1, \ldots, 4n,$$

$$R_j^{\mathcal{A}} := \big\{ \langle \mathsf{a}_{S_j}^{(j)}, \ldots, \mathsf{a}_{S_n}^{(n)} \rangle \in \mathsf{A}^n \;\big|\; S_j \in S_2 \in \ldots \in S_n \big\} \text{ for } j = 1, \ldots, n, \text{ and}$$

$$T_j^{\mathcal{A}} := \big\{ \langle \mathsf{b}_{S_j}^{(j)}, \ldots, \mathsf{b}_{S_n}^{(n)} \rangle \in \mathsf{A}^n \;\big|\; S_j \in S_2 \in \ldots \in S_n \big\} \text{ for } j = 1, \ldots, n.$$

For any choice of $S_1, \ldots, S_n$ with $S_1 \in \ldots \in S_n$ we observe

$$\mathcal{A}, \big[x_1 \mapsto \mathsf{a}_{S_1}^{(1)}, \ldots, x_n \mapsto \mathsf{a}_{S_n}^{(n)}\big] \models \bigwedge_{j=1}^{n} R_j(x_j, \ldots, x_n) \;,$$

$$\mathcal{A}, \big[y_1 \mapsto \mathsf{b}_{S_1}^{(1)}, \ldots, y_n \mapsto \mathsf{b}_{S_n}^{(n)}\big] \models \bigwedge_{j=1}^{n} T_j(y_j, \ldots, y_n) \;, \text{ and}$$

$$\mathcal{A}, \big[x_1 \mapsto \mathsf{a}_{S_1}^{(1)}, \ldots, x_n \mapsto \mathsf{a}_{S_n}^{(n)}, y_1 \mapsto \mathsf{b}_{S_1}^{(1)}, \ldots, y_n \mapsto \mathsf{b}_{S_n}^{(n)}\big] \models \bigwedge_{i=1}^{4n} P_i(x_1, \ldots, x_n) \leftrightarrow Q_i(y_1, \ldots, y_n) \;.$$

For any tuple $\langle \mathsf{c}_j, \ldots, \mathsf{c}_n \rangle$ for which there do not exist sets $S_j \in \mathcal{S}_j, \ldots, S_n \in \mathcal{S}_n$ such that $S_j \in \ldots \in S_n$ and $\langle \mathsf{c}_1, \ldots, \mathsf{c}_n \rangle$ equals $\langle \mathsf{a}_{S_j}^{(j)}, \ldots, \mathsf{a}_{S_n}^{(n)} \rangle$, we observe $\mathcal{A}, [x_j \mapsto \mathsf{c}_j, \ldots, x_n \mapsto \mathsf{c}_n] \not\models R_j(x_j, \ldots, x_n)$. Hence, for any variable assignment β we have

$$\mathcal{A}, \beta[x_j \mapsto \mathsf{c}_j, \ldots, x_n \mapsto \mathsf{c}_n] \models R_j(x_j, \ldots, x_n) \rightarrow \ldots \;.$$

Consequently, $\mathcal{A}$ is a model of φ.

$\mathfrak{A}, \mathfrak{B}$

Consider the following simple two-player game with Players $\mathfrak{A}$ and $\mathfrak{B}$ where both players have complete and instantaneous knowledge about all moves that are made by either player. In the first round $\mathfrak{A}$ moves first by picking some domain element $\mathsf{a}_{S_{\mathfrak{A},n}}^{(n)}$ for some set $S_{\mathfrak{A},n} \in \mathcal{S}_n$. $\mathfrak{B}$ knows about $\mathfrak{A}$'s choice and answers by picking a domain element $\mathsf{b}_{S_{\mathfrak{B},n}}^{(n)}$ for some set $S_{\mathfrak{B},n} \in \mathcal{S}_n$. The game continues for $n-1$ more rounds, where in every round Player $\mathfrak{A}$ picks a domain element $\mathsf{a}_{S_{\mathfrak{A},j}}^{(j)}$ with $S_{\mathfrak{A},j} \in S_{\mathfrak{A},j+1}$ and $\mathfrak{B}$ answers by picking some $\mathsf{b}_{S_{\mathfrak{B},j}}^{(j)}$ with $S_{\mathfrak{B},j} \in S_{\mathfrak{B},j+1}$. Hence, in the last round the chosen domain elements $\mathsf{a}_{S_{\mathfrak{A},1}}^{(1)}$ and $\mathsf{b}_{S_{\mathfrak{B},1}}^{(1)}$ are such that $S_{\mathfrak{A},1}$ and $S_{\mathfrak{B},1}$ are both nonempty subsets of $[4n]$. Player $\mathfrak{A}$ wins if and only if

$S_{\mathfrak{A},j}$, $\mathsf{a}_{S_{\mathfrak{A},j}}^{(j)}$
$S_{\mathfrak{B},j}$, $\mathsf{b}_{S_{\mathfrak{B},j}}^{(j)}$

$$\mathcal{A}, \big[x_1 \mapsto \mathsf{a}_{S_{\mathfrak{A},1}}^{(1)}, \ldots, x_n \mapsto \mathsf{a}_{S_{\mathfrak{A},n}}^{(n)}, y_1 \mapsto \mathsf{b}_{S_{\mathfrak{B},1}}^{(1)}, \ldots, y_n \mapsto \mathsf{b}_{S_{\mathfrak{B},n}}^{(n)}\big] \not\models P_i(x_1, \ldots, x_n) \leftrightarrow Q_i(y_1, \ldots, y_n)$$

for some $i \in [4n]$, and Player $\mathfrak{B}$ wins if and only if

$$\mathcal{A}, \big[x_1 \mapsto \mathsf{a}_{S_{\mathfrak{A},1}}^{(1)}, \ldots, x_n \mapsto \mathsf{a}_{S_{\mathfrak{A},n}}^{(n)}, y_1 \mapsto \mathsf{b}_{S_{\mathfrak{B},1}}^{(1)}, \ldots, y_n \mapsto \mathsf{b}_{S_{\mathfrak{B},n}}^{(n)}\big] \models P_i(x_1, \ldots, x_n) \leftrightarrow Q_i(y_1, \ldots, y_n)$$

for every $i \in [4n]$. Since $\mathcal{A}$ is a model of φ, there must exist a winning strategy for $\mathfrak{B}$.

Claim I: There is exactly one winning strategy for $\mathfrak{B}$, namely, for every $j = n, \ldots, 1$ Player $\mathfrak{B}$ picks the element $\mathsf{b}^{(j)}_{S_{\mathfrak{A},j}}$ in round $n - j + 1$, i.e. for every j we have $S_{\mathfrak{B},j} = S_{\mathfrak{A},j}$.

Proof: It is easy to see that the described strategy is a winning strategy for $\mathfrak{B}$.

Assume $\mathfrak{B}$ deviates from this strategy. This means there exists some j_*, $1 \leq j_* \leq n$, such that $\mathfrak{B}$ did not adhere to the described strategy in the $(n - j_* + 1)$-st round, i.e. $S_{\mathfrak{B},j_*} \neq S_{\mathfrak{A},j_*}$.

We show by induction on j_* that $\mathfrak{A}$ has a winning strategy from this deviation point on.

For the base case $j_* = 1$ we consider two distinct nonempty sets $S_{\mathfrak{A},1}, S_{\mathfrak{B},1} \subseteq [4n]$. There must be some index i_* that belongs to one of the two sets but not to the other, i.e. $i_* \in (S_{\mathfrak{A},1} \cup S_{\mathfrak{B},1}) \setminus (S_{\mathfrak{A},1} \cap S_{\mathfrak{B},1})$.

Suppose that $i_* \in S_{\mathfrak{A},1} \setminus S_{\mathfrak{B},1}$. Hence, we can construct the chain $i_* \in S_{\mathfrak{A},1} \in \ldots \in S_{\mathfrak{A},n}$, by definition of the allowed moves. This entails $\mathcal{A}, [x_1 \mapsto \mathsf{a}^{(1)}_{S_{\mathfrak{A},1}}, \ldots, x_n \mapsto \mathsf{a}^{(n)}_{S_{\mathfrak{A},n}}] \models P_{i_*}(x_1, \ldots, x_n)$. On the other hand, we get $\mathcal{A}, [y_1 \mapsto \mathsf{b}^{(1)}_{S_{\mathfrak{B},1}}, \ldots, y_n \mapsto \mathsf{b}^{(n)}_{S_{\mathfrak{B},n}}] \not\models Q_{i_*}(y_1, \ldots, y_n)$, because of $i_* \notin S_{\mathfrak{B},1}$. Hence, $\mathfrak{A}$ wins and the chosen strategy cannot be a winning strategy for $\mathfrak{B}$.

The case where $i_* \in S_{\mathfrak{B},1} \setminus S_{\mathfrak{A},1}$ is symmetric and $\mathfrak{A}$ wins as well.

For the inductive case we fix some $j_* > 1$. Since $S_{\mathfrak{A},j_*}$ and $S_{\mathfrak{B},j_*}$ are distinct but have the same number of elements, there is some set $S' \in S_{\mathfrak{A},j_*} \setminus S_{\mathfrak{B},j_*}$. If $\mathfrak{A}$ picks $\mathsf{a}^{(j_*-1)}_{S_{\mathfrak{A},j_*-1}} := \mathsf{a}^{(j_*-1)}_{S'}$ in the following round, we have $S_{\mathfrak{B},j_*-1} \neq S_{\mathfrak{A},j_*-1}$ for any choice $\mathsf{b}^{(j_*-1)}_{S_{\mathfrak{B},j_*-1}}$ that $\mathfrak{B}$ could possibly make in accordance with the rules. By induction, $\mathfrak{A}$ has a winning strategy starting from the next round of the game. Hence, there is a winning strategy for $\mathfrak{A}$ starting from the current round. ◊

The described game corresponds to the model-checking game associated with the pair $\langle \mathcal{A}, \varphi \rangle$. Obviously, the given rules limit the moves of the involved players in such a way that all guards in φ are satisfied by the variable assignment both players construct move by move. Viewed in this light, the above claim proves the following observation. For every $S \in \mathcal{S}_k$, $1 \leq k \leq n$, we define the structure $\mathcal{A}_{-S}$ as the substructure of $\mathcal{A}$ induced by the domain $\mathsf{A}_{-S} := \mathsf{A} \setminus \{\mathsf{b}^{(k)}_S\}$. $\mathcal{A}_{-S}$

Claim II: For every $S \in \mathcal{S}_k$, $1 \leq k \leq n$, the substructure $\mathcal{A}_{-S}$ of $\mathcal{A}$ does not satisfy φ.

Proof: The reason is simply that in this case player $\mathfrak{A}$ can always prevent $\mathfrak{B}$ from reaching a state of the game where $\mathfrak{B}$ can apply the described winning strategy. ◊

We have already analyzed the size of the sets $\mathcal{S}_k$. Due to the observed lower bounds, we know that A contains at least $\sum_{k=1}^{n} 2^{\uparrow k}(n)$ elements of the form $\mathsf{b}^{(k)}_S$.

Let φ_{LGF} be a shortest LGF sentence that is semantically equivalent to φ. Next, we argue that $\mathrm{len}(\varphi_{\mathrm{LGF}})$ is at least $(n-1)$-fold exponential in n. We start by introducing some additional notation. We divide the domain A into two disjoint parts $\mathsf{A}_\mathsf{a} := \{\mathsf{a}^{(k)}_S \mid 1 \leq k \leq n \text{ and } S \in \mathcal{S}_k\}$ and $\mathsf{A}_\mathsf{b} := \{\mathsf{b}^{(k)}_S \mid 1 \leq k \leq n \text{ and } S \in \mathcal{S}_k\}$. Moreover, we subdivide A_b into parts $\mathsf{A}_{\mathsf{b},k} := \{\mathsf{b}^{(k)}_S \mid S \in \mathcal{S}_k\}$ with $1 \leq k \leq n$. We define the following vocabularies A_a, A_b, $\mathsf{A}_{\mathsf{b},k}$

$$\begin{aligned} \Sigma &:= \langle \{P_i, Q_i \mid 1 \leq i \leq 4n\} \cup \{R_j, T_j \mid 1 \leq j \leq n\}, \emptyset \rangle \,, \\ \Sigma_{PR} &:= \langle \{P_i \mid 1 \leq i \leq 4n\} \cup \{R_j \mid 1 \leq j \leq n\}, \emptyset \rangle \,, \text{ and} \\ \Sigma_{QT} &:= \langle \{Q_i \mid 1 \leq i \leq 4n\} \cup \{T_j \mid 1 \leq j \leq n\}, \emptyset \rangle \,. \end{aligned}$$

$\Sigma, \Sigma_{PR}, \Sigma_{QT}$

Moreover, let Σ'_{PR} and Σ'_{QT} be disjoint extensions of the vocabularies Σ_{PR} and Σ_{QT}, respectively, each extended by a countably infinite number of nullary predicate symbols. Σ'_{PR}, Σ'_{QT}

Claim III-a: Consider any loose guard $\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ over the vocabulary Σ. Suppose we have $\mathcal{A}, \beta \models \gamma$ for some variable assignment β over $\mathcal{A}$'s domain, or $\mathcal{A}_{-S}, \beta \models \gamma$ for some S and some variable assignment β over the domain of $\mathcal{A}_{-S}$. Then, either all atoms in γ are Σ_{PR}-atoms or all atoms in γ are Σ_{QT}-atoms.

Proof: We argue for the case where $\mathcal{A}, \beta \models \gamma$. The argument for the cases $\mathcal{A}_{-S}, \beta \models \gamma$ are the same.

Let $\gamma(\bar{u}, \bar{v}) = A_1(\bar{u}, \bar{v}) \wedge \ldots \wedge A_m(\bar{u}, \bar{v})$. Suppose there are k, k' such that A_k is a Σ_{PR}-atom and $A_{k'}$ is a Σ_{QT}-atom. Let $V := \mathrm{vars}(A_k)$ and $V' := \mathrm{vars}(A_{k'})$. Because of $\mathcal{A}, \beta \models A_k \wedge A_{k'}$, we must have $\beta(V) \subseteq \mathsf{A_a}$ and $\beta(V') \subseteq \mathsf{A_b}$. Since γ is a loose guard, there are variables $u \in \bar{u}$ and $v \in V$ and $v' \in V'$ and atoms A_ℓ, A'_ℓ such that u co-occurs with v in A_ℓ and u co-occurs with v' in $A_{\ell'}$. Because of $\mathcal{A}, \beta \models A_\ell \wedge A_{\ell'}$, we must have $\beta\big(\mathrm{vars}(A_\ell)\big) \subseteq \mathsf{A_a}$ and $\big(\mathrm{vars}(A_\ell)\big) \subseteq \mathsf{A_b}$. Hence, $\beta(u) \in \mathsf{A_a} \cap \mathsf{A_b}$. But this contradicts the fact that $\mathsf{A_a}$ and $\mathsf{A_b}$ are disjoint.

Consequently, γ must either be a conjunction of Σ_{PR}-atoms or a conjunction of Σ_{QT}-atoms, but cannot mix the two kinds. ◊

Claim III-b: Consider any loosely guarded Σ-formula $\chi(\bar{v}) := \big(\mathcal{Q}\bar{u}.\, \gamma(\bar{u}, \bar{v})\big)\eta(\bar{u}, \bar{v})$ and let $A(\bar{z})$ be some atom occurring in $\eta(\bar{u}, \bar{v})$. Recall that we assume that none of the variables occurring freely in any subformula of χ occurs bound in the same subformula and that, moreover, no variable is bound by two distinct occurrences of quantifiers. Suppose that $\gamma(\bar{u}, \bar{v})$ is a Σ_{PR}-formula and that $A(\bar{z})$ is a Σ_{QT}-atom or vice versa. Further suppose that we have $\mathcal{A}, \beta \models \gamma(\bar{u}, \bar{v})$ for some variable assignment β over $\mathcal{A}$'s domain. Let Z be the set of variables occurring in $A(\bar{z})$ that are free in $\eta(\bar{u}, \bar{v})$. Either Z is empty, i.e. none of the variables in $A(\bar{z})$ occurs freely in $\eta(\bar{u}, \bar{v})$, or we have $\mathcal{A}, \beta' \not\models A(\bar{z})$ for every variable assignment over $\mathcal{A}$'s domain that coincides with β on the variables in Z.

The same holds if we replace $\mathcal{A}$ by any $\mathcal{A}_{-S}$.

Proof: We treat the case where γ is a Σ_{PR}-formula and A is a Σ_{QT}-atom. The other case can be treated in a similar way. Consider some β with $\mathcal{A}, \beta \models \gamma(\bar{u}, \bar{v})$ and any β' that coincides with β on Z and which satisfies $\mathcal{A}, \beta' \models A(\bar{z})$. Then, we observe $\beta(\bar{u} \cup \bar{v}) \subseteq \mathsf{A_a}$ and $\beta'(Z) \subseteq \mathsf{A_b}$. This entails $Z \cap (\bar{u} \cup \bar{v}) = \emptyset$, as $\mathsf{A_a}$ and $\mathsf{A_b}$ are disjoint. ◊

Claim III-c: Consider any loosely guarded Σ-formula $\chi(\bar{v}) := \big(\mathcal{Q}\bar{u}.\, \gamma(\bar{u}, \bar{v})\big)\chi'(\bar{u}, \bar{v})$ where $\chi'(\bar{u}, \bar{v})$ contains a subformula $\eta(\bar{z})$ of the form $\big(\mathcal{Q}'\bar{y}.\, \delta(\bar{y}, \bar{z})\big)\eta'(\bar{y}, \bar{z})$. Moreover, we assume that the guard $\gamma(\bar{u}, \bar{v})$ is a Σ_{PR}-formula and the guard $\delta(\bar{y}, \bar{z})$ is a Σ_{QT}-formula or vice versa.

Suppose we have $\mathcal{A}, \beta \models \gamma(\bar{u}, \bar{v})$ for some variable assignment β over $\mathcal{A}$'s domain. Let Z be the set variables occurring in $\eta'(\bar{y}, \bar{z})$ that occur freely in $\chi'(\bar{u}, \bar{v})$, i.e. $Z \subseteq \bar{z}$. Let β' be any variable assignment that coincides with β on Z. Then, either Z is empty or $\mathcal{A}, \beta' \not\models \delta(\bar{y}, \bar{z})$.

The same holds if we replace $\mathcal{A}$ by any $\mathcal{A}_{-S}$.

Proof: By definition of LGF formulas, for every $z \in \bar{z}$ that occurs in η' and is free in χ' the guard $\delta(\bar{y}, \bar{z})$ must contain at least one atom with z as argument. Consequently, the claim follows from Claim III-b. ◊

Starting from φ_{LGF} we construct the sentence ψ_{LGF} which possesses the following properties.

(a) The sentence ψ_{LGF} is a Boolean combination of loosely-guarded Σ'_{PR}-sentences and loosely-guarded Σ'_{QT}-sentences. Moreover, ψ_{LGF} is in negation normal form.

(b) The vocabulary underlying ψ_{LGF} is that of φ_{LGF} extended by fresh nullary predicate symbols.

$\mathcal{B}$, $\mathcal{B}_{-S}$ (c) The structure $\mathcal{A}$ can be uniquely expanded to a model $\mathcal{B}$ of ψ_{LGF} over the same domain and conserving the interpretations of all predicate symbols occurring in φ_{LGF}; for every $\mathcal{B}_{-S}$ — which is defined to be the substructure of $\mathcal{B}$ induced by the domain $\mathsf{B}_{-S} := \mathsf{B} \setminus \{\mathsf{b}_S^{(k)}\}$ for any $S \in \mathcal{S}_k$ — we have $\mathcal{B}_{-S} \not\models \psi_{\mathrm{LGF}}$.

(d) $\mathrm{len}(\psi_{\mathrm{LGF}}) \in \mathcal{O}\big(\mathrm{len}(\varphi_{\mathrm{LGF}})\big)$.

The construction of ψ_{LGF} starts from φ_{LGF}. As a first step, we shift any negation signs in φ_{LGF} inwards so that they occur directly in front of atoms. We do this in a way that preserves guarded quantification. The length of the resulting formula is linear in the length of φ_{LGF}. Next, we perform the following steps.

(1) Remove any subformulas that have the form $\big(\mathcal{Q}\bar{\mathrm{u}}.\,\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})\big)\chi$ where γ is neither a Σ_{PR}-formula nor a Σ_{QT}-formula but mixes Σ_{PR}- and Σ_{QT}-atoms. Since, by Claim III-a, $\mathcal{A},\beta \not\models \gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$ and $\mathcal{A}_{-S},\beta \not\models \gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$ for every β and every S, these subformulas can be replaced by `true` in case of $\mathcal{Q} = \forall$ and by `false` if $\mathcal{Q} = \exists$.

All guards in the resulting formula are either Σ_{PR}-formulas or Σ_{QT}-formulas.

(2) Remove any subformulas η of the form $\big(\mathcal{Q}'\bar{\mathrm{y}}.\,\delta(\bar{\mathrm{y}},\bar{\mathrm{z}})\big)\eta'(\bar{\mathrm{y}},\bar{\mathrm{z}})$ that occur in subformulas $\chi := \big(\mathcal{Q}\bar{\mathrm{u}}.\,\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})\big)\chi'$ where γ is a Σ_{PR}-guard and δ is a Σ_{QT}-guard or vice versa, and where $\eta'(\bar{\mathrm{y}},\bar{\mathrm{z}})$ contains variables that are free in χ'. By Claim III-c, $\mathcal{A},\beta \models \gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$ entails $\mathcal{A},\beta' \not\models \delta(\bar{\mathrm{y}},\bar{\mathrm{z}})$ for every β and every β' which coincides with β on all variables that occur freely in χ'. Hence, in case of $\mathcal{Q}' = \forall$ we can replace η with `true`, and in case of $\mathcal{Q}' = \exists$ we can replace η with `false`.

(3) Remove any Σ_{QT}-atom (Σ_{PR}-atom) $A(\bar{\mathrm{z}})$ that lies in the scope χ of a Σ_{PR}-guarded (Σ_{QT}-guarded) quantified subformula $\big(\mathcal{Q}\bar{\mathrm{u}}.\,\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})\big)\chi$ and in which a variable $z \in \bar{\mathrm{z}} \cap (\bar{\mathrm{u}} \cup \bar{\mathrm{v}})$ occurs. By Claim III-b, $\mathcal{A},\beta \models \gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$ entails $\mathcal{A},\beta' \not\models A(\bar{\mathrm{z}})$ for every β and every β' which coincides with β on all variables that occur freely in χ. Hence, we can replace $A(\bar{\mathrm{z}})$ with `false`.

(4) Do the following steps iteratively and exhaustively. Replace every occurrence of a non-atomic Σ'_{PR}-sentence (Σ'_{QT}-sentence) χ in φ_{LGF} which does not contain another non-atomic sentence as proper subformula with the atom M, where M is a fresh nullary predicate symbol, and conjoin the formula $M \leftrightarrow \chi$. We take M from Σ'_{PR} if the smallest quantifier scope the replaced occurrence of χ belongs to is Σ_{PR}-guarded. Otherwise, we take some nullary M from Σ'_{QT}. The resulting formula is $\varphi_{\mathrm{LGF}}[\chi/M] \wedge (M \leftrightarrow \chi)$.

The final result of this process has a length that is linear in the length of the original. Due to the previous transformations, we obtain a sentence that is a Boolean combination of Σ'_{PR}-sentences and Σ'_{QT}-sentences and satisfies Properties (a) to (a). We shall call it ψ_{LGF} from now on. Moreover, none of the constituent sentences of ψ_{LGF} properly contains a non-atomic sentence.

We need some more notions and notation. An atom is called *linear* if every variable in it occurs at most once. Any occurrence of a variable v in a non-equational Σ-atom A is called a *column-k-occurrence*, if v is the $(n-k+1)$-st argument from the right in A. For example, if we fix n to be 6, then v has a column-5-occurrence in each of the atoms $Q_i(x_1,x_2,x_3,x_4,v,x_6)$, $T_3(x_3,x_4,v,x_6)$, $T_5(v,x_6)$, but v has no column-5-occurrence in the atoms $T_6(x_6)$ or $Q_i(v,v,v,v,x_5,v)$.

linear atoms

column-k-occurrences

Claim IV-a: Let $A(\bar{\mathrm{v}})$ be some non-equational Σ_{QT}-atom. Consider any variable v that has a column-k-occurrence in $A(\bar{\mathrm{v}})$. Then, for every variable assignment β we observe that $\mathcal{B},\beta \models A(\bar{\mathrm{v}})$ entails $\beta(v) \in \mathsf{A}_{\mathsf{b},k}$. Similarly, for every S we have that $\mathcal{B}_{-S},\beta \models A(\bar{\mathrm{v}})$ entails $\beta(v) \in \mathsf{A}_{\mathsf{b},k}$.

The same holds if we replace $\mathcal{B}$ with $\mathcal{B}_{-S}$ for any S.

Proof: This follows immediately from the definition of $\mathcal{B}$. ◊

Claim IV-b: Consider any loose guard $\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$ over the vocabulary Σ'_{QT}. If $\mathcal{B},\beta \models \gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$ or $\mathcal{B}_{-S},\beta \models \gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$ holds for some variable assignment β and any S, then for every variable v occurring in a non-equational atom in $\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$ there is a unique k, $1 \leq k \leq n$, such that every occurrence of v in a non-equational atom in $\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$ is a column-k-occurrence.

The same holds if we replace $\mathcal{B}$ with $\mathcal{B}_{-S}$ for any S.

Proof: Suppose v has a column-k-occurrence in $\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$ and, at the same time, a column-k'-occurrence in $\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$ with $k \neq k'$. By Claim IV-a, we then have $\beta(v) \in \mathsf{A}_{\mathsf{b},k} \cap \mathsf{A}_{\mathsf{b},k'}$. But since this intersection is empty, we obtain a contradiction. $\Diamond$

Claim IV-c: Consider any loosely guarded Σ_{QT}-formula $\chi(\bar{\mathrm{v}}) := \big(\mathcal{Q}\bar{\mathrm{u}}.\,\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})\big)\eta(\bar{\mathrm{u}},\bar{\mathrm{v}})$ in which $\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$ contains at least one non-equational atom. Suppose there is a maximal subset $Z \subseteq \bar{\mathrm{u}}\cup\bar{\mathrm{v}}$ containing at least two distinct variables such that $\mathcal{B},\beta \models \gamma(\bar{\mathrm{u}},\bar{\mathrm{v}}) \rightarrow \bigwedge_{z,z'\in Z} z \approx z'$ holds for every variable assignment β. Then, there is some variable $z_* \in Z$ that occurs in some non-equational atom $A(\bar{\mathrm{z}})$ in $\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$. Moreover, $\chi(\bar{\mathrm{v}})$ is equivalent to some loosely-guarded Σ_{QT}-formula $\big(\mathcal{Q}\bar{\mathrm{u}}'.\,\gamma'(\bar{\mathrm{u}}',\bar{\mathrm{v}}')\big)\eta'(\bar{\mathrm{u}}',\bar{\mathrm{v}}')$, where $z_1,\ldots,z_m$ is an enumeration of all the variables in Z, $\gamma'(\bar{\mathrm{u}}',\bar{\mathrm{v}}')$ is the result of removing any trivial equations from $\gamma\big[z_1/z_*,\ldots,z_m/z_*\big]$, and $\eta' := \eta\big[z_1/z_*,\ldots,z_m/z_*\big]$.

The same holds if we replace $\mathcal{B}$ with $\mathcal{B}_{-S}$ for any S.

Proof: Suppose that none of the $z \in Z$ occurs in any non-equational atom in $\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$.

Consider some $u \in \bar{\mathrm{u}}$ that does not occur in any non-equational atom in $\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$. Then, u must co-occur with every $z \in Z$ in some equation in $\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$. But then, we have $\mathcal{B},\beta \models \gamma(\bar{\mathrm{u}},\bar{\mathrm{v}}) \rightarrow u \approx z$ for every $z \in Z$. Since Z is maximal, we get $u \in Z$. Moreover, every v that occurs in some non-equational atom in $\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$ must co-occur with u in some equation in $\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$. Again, this entails $v \in Z$ and, hence, yields a contradiction.

Consider some $u \in \bar{\mathrm{u}}$ that occurs in some non-equational atom in $\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$. Then, u must co-occur with every $z \in Z$ in some equation in $\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$. But then, we once more get $u \in Z$.

Consequently, there is some $z_* \in Z$ that occurs in some non-equational atom $A(\bar{\mathrm{z}})$ in $\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$. The rest of the claim follows immediately. $\Diamond$

Claim IV-d: Consider any loosely guarded Σ_{QT}-sentence $\chi := \big(\mathcal{Q}\bar{\mathrm{u}}.\,\gamma(\bar{\mathrm{u}},\emptyset)\big)\eta(\bar{\mathrm{u}})$ in which $\gamma(\bar{\mathrm{u}},\emptyset)$ contains exclusively equational atoms. Let u be some variable from $\bar{\mathrm{u}}$ and let $u_1,\ldots,u_m$ be an enumeration of all the variables occurring from $\bar{\mathrm{u}}$. Then, χ is equivalent to the sentence $\big(\mathcal{Q}u.\,u \approx u\big)\eta'(u)$ where $\eta' := \eta\big[u_1/u,\ldots,u_m/u\big]$. Moreover, for any atom $A(u)$ in $\eta'(u)$ that does not lie within the scope of any quantifier in $\eta'(u)$ we have that either $A(u)$ is a trivial equation $u \approx u$, or $A(u)$ is of the form $T_n(u)$, or $\mathcal{B},\beta \not\models A(u)$ for every variable assignment β.

The same holds if we replace $\mathcal{B}$ with $\mathcal{B}_{-S}$ for any S.

Proof: Since every variable occurring in $\gamma(\bar{\mathrm{u}})$ must co-occur with u in some equation in $\gamma(\bar{\mathrm{u}})$, we have $\mathcal{B},\beta \models \gamma(\bar{\mathrm{u}}) \rightarrow \bigwedge_{v\in\bar{\mathrm{u}}} u \approx v$ for every variable assignment β. Hence, χ is equivalent to $\big(\mathcal{Q}u.\,u \approx u\big)\eta'(u)$. Since we assume $n \geq 3$, every predicate symbol Q_i has arity 3. Hence, any non-equational atom $A(u)$ with any predicate symbol from Σ_{QT} different from T_n must contain more than one occurrence of u. But in such cases we get $\mathcal{B},\beta \not\models A(u)$ for every variable assignment β. $\Diamond$

Claim IV-e: Consider any loosely guarded Σ_{QT}-formula $\chi(\bar{\mathrm{v}}) := \big(\mathcal{Q}\bar{\mathrm{u}}.\,\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})\big)\eta(\bar{\mathrm{u}},\bar{\mathrm{v}})$ in which $\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$ contains exclusively equational atoms and $\bar{\mathrm{v}}$ is not empty. Let v_* be some variable from $\bar{\mathrm{v}}$ and let $z_1,\ldots,z_m$ be an enumeration of all the variables occurring in $\bar{\mathrm{u}}\cup\bar{\mathrm{v}}$. Then, $\chi(\bar{\mathrm{v}})$ is equivalent to the formula $\big(\bigwedge_{v\in\bar{\mathrm{v}}\cap\mathrm{vars}(\gamma)} v \approx v_*\big) \rightarrow \eta'(v_*)$, if $\mathcal{Q} = \forall$, and $\chi(\bar{\mathrm{v}})$ is equivalent to the formula $\big(\bigwedge_{v\in\bar{\mathrm{v}}\cap\mathrm{vars}(\gamma)} v \approx v_*\big) \wedge \eta'(v_*)$, if $\mathcal{Q} = \exists$, where $\eta' := \eta\big[z_1/v_*,\ldots,z_m/v_*\big]$. Moreover, for any atom $A(v_*)$ in $\eta'(v_*)$ that does not lie within the scope of any quantifier in $\eta'(v_*)$ we have that either $A(v_*)$ is a trivial equation $v_* \approx v_*$, or $A(v_*)$ is of the form $T_n(v_*)$, or $\mathcal{B},\beta \not\models A(v_*)$ for every variable assignment β.

The same holds if we replace $\mathcal{B}$ with $\mathcal{B}_{-S}$ for any S.

Proof: Pick some variable $u_* \in \bar{\mathrm{u}}$. Since every variable occurring in $\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$ must co-occur with u_* in some equation in $\gamma(\bar{\mathrm{u}},\bar{\mathrm{v}})$, we have $\mathcal{B},\beta \models \gamma(\bar{\mathrm{u}},\bar{\mathrm{v}}) \rightarrow \bigwedge_{z\in\bar{\mathrm{u}}\cup\bar{\mathrm{v}}} u_* \approx z$ for every variable

assignment β. Hence, $\chi(\bar{\mathrm{v}})$ is equivalent to the formula $\big(\bigwedge_{v\in\bar{\mathrm{v}}\cap\mathrm{vars}(\gamma)} v \approx v_*\big) \to \eta'(v_*)$, if $\mathcal{Q} = \forall$, and $\chi(\bar{\mathrm{v}})$ is equivalent to the formula $\big(\bigwedge_{v\in\bar{\mathrm{v}}\cap\mathrm{vars}(\gamma)} v \approx v_*\big) \wedge \eta'(v_*)$, if $\mathcal{Q} = \exists$.

Since we assume $n \geq 3$, every predicate symbol Q_i has arity 3. Hence, any atom $A(u)$ with any predicate symbol from Σ_{QT} different from T_n must contain more than one occurrence of u. But in such cases we get $\mathcal{B}, \beta \not\models A(u)$ for every variable assignment β. ◊

Claim IV-f: Consider any loosely guarded Σ'_{QT}-formula $\chi(\bar{\mathrm{v}}) := \big(\mathcal{Q}\bar{\mathrm{u}}.\, \gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})\big)\eta(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ and let $A(\bar{\mathrm{z}})$ be some non-equational Σ_{QT}-atom occurring in $\eta(\bar{\mathrm{u}}, \bar{\mathrm{v}})$. Suppose that there is some variable v with a column-k-occurrence in some non-equational atom in $\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ and with a column-k'-occurrence in $A(\bar{\mathrm{z}})$ for distinct k, k'. For every β we observe that $\mathcal{B}, \beta \models \gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ entails $\mathcal{B}, \beta' \not\models A(\bar{\mathrm{z}})$ for every variable assignment β' over $\mathcal{B}$'s domain that coincides with β on the variables that occur freely in $\eta(\bar{\mathrm{u}}, \bar{\mathrm{v}})$.

The same holds if we replace $\mathcal{B}$ with $\mathcal{B}_{-S}$ for any S.

Proof: As above, $\mathcal{A}, \beta \models \gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ together with $\mathcal{A}, \beta' \models A(\bar{\mathrm{z}})$ entails $\beta(v) = \beta'(v) \in \mathsf{A}_{\mathsf{b},k} \cap \mathsf{A}_{\mathsf{b},k'}$ for any $v \in \bar{\mathrm{z}}$ that occurs freely in $\eta(\bar{\mathrm{u}}, \bar{\mathrm{v}})$. However, this intersection yields an empty set. ◊

Claim IV-g: Consider any loose guard $\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ over the vocabulary Σ'_{QT}. Let $u \in \bar{\mathrm{u}}$ be a variable that has a column-k-occurrence in $\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$. If $\mathcal{B}, \beta \models \gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ or $\mathcal{B}_{-S}, \beta \models \gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ holds for some variable assignment β and any S, then there is no variable $v \neq u$ with column-k-occurrences in γ, unless $\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ contains an equation $u \approx v$.

The same holds if we replace $\mathcal{B}$ with $\mathcal{B}_{-S}$ for any S.

Proof: For every variable v occurring in $\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ the guard $\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ must contain some atom $A(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ in which u and v co-occur. If $A(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ is not an equation, then, by Claim IV-b, the occurrence of u in $A(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ is a column-k-occurrence. Hence, the (unique) occurrence of v in $A(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ is a column-k'-occurrence for some $k' \neq k$. By Claim IV-b, all occurrences of v in any non-equational atom in $\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ are column-k'-occurrences. ◊

Claim IV-h: Consider any loosely guarded Σ'_{QT}-formula $\chi(\bar{\mathrm{v}}) := \big(\mathcal{Q}\bar{\mathrm{u}}.\, \gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})\big)\eta(\bar{\mathrm{u}}, \bar{\mathrm{v}})$. Let $u \in \bar{\mathrm{u}}$ be a variable that has a column-k-occurrence in $\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$. Suppose that we have $\mathcal{B}, \beta \models \gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ for some variable assignment β over $\mathcal{B}$'s domain. Further suppose that $\eta(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ contains a non-equational Σ_{QT}-atom $A(\bar{\mathrm{z}})$ in which some variable v has a column-k-occurrence. If v occurs freely in $\eta(\bar{\mathrm{u}}, \bar{\mathrm{v}})$, then we either have $v = u$ or $\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ contains the equation $u \approx v$ or $\mathcal{B}, \beta' \not\models A(\bar{\mathrm{z}})$ for every variable assignment β' over $\mathcal{B}$'s domain that coincides with β on the variables that occur freely in $\eta(\bar{\mathrm{u}}, \bar{\mathrm{v}})$.

The same holds if we replace $\mathcal{B}$ with $\mathcal{B}_{-S}$ for any S.

Proof: Suppose $v \neq u$ and that $\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ does not contain the equation $u \approx v$. Since v occurs freely in $\eta(\bar{\mathrm{u}}, \bar{\mathrm{v}})$, it must also occur in some non-equational atom in $\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$. By Claims IV-g and IV-b, v has only column-k'-occurrences in $\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ with $k' \neq k$. Hence, Claim IV-f entails $\mathcal{B}, \beta' \not\models A(\bar{\mathrm{z}})$ for every variable assignment β' over $\mathcal{B}$'s domain that coincides with β on the variables that occur freely in $\eta(\bar{\mathrm{u}}, \bar{\mathrm{v}})$. ◊

Due to Claims IV-c to IV-e, we can reduce the equations occurring in guards in Σ'_{QT}-subformulas of ψ_{LGF} to a minimum without losing properties (a) – (d). The only equations in guards that cannot be removed in this way are part of purely equational guards that belong to top-most quantifiers in Σ'_{QT}-sentences. These guards consist of exactly one trivial equation.

Due to Claims IV-f and IV-h, we can modify ψ_{LGF} as follows while retaining properties (a)–(d). Let $\big(\mathcal{Q}\bar{\mathrm{u}}.\, \gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})\big)\eta(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ be any Σ'_{QT}-guarded subformula of ψ_{LGF}. For any variable v that has a column-k-occurrence in $\gamma(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ we can replace any atom $A(\bar{\mathrm{z}})$ in $\eta(\bar{\mathrm{u}}, \bar{\mathrm{v}})$ in which v has a column-k'-occurrence with $k' \neq k$ by `false`. If this occurrence of $A(\bar{\mathrm{z}})$ is part of a guard $\delta(\bar{\mathrm{x}}, \bar{\mathrm{y}})$ of a subformula $\big(\mathcal{Q}\bar{\mathrm{x}}.\, \delta(\bar{\mathrm{x}}, \bar{\mathrm{y}})\big)\eta'(\bar{\mathrm{x}}, \bar{\mathrm{y}})$, then we replace the whole subformula with `true` if $\mathcal{Q} = \forall$, and we replace the whole subformula with `false` if $\mathcal{Q} = \exists$. We proceed analogously for atoms $A(\bar{\mathrm{z}})$ containing

column-k-occurrences of variables v' with $v' \neq v$. In the resulting formula, every non-equational Σ_{QT}-atom is linear and for every variable v occurring in any non-equational Σ_{QT}-atom there is some k such that all occurrences of v in non-equational $\Sigma_{Q,T}$-atoms are column-k-occurrences. Moreover, for every $\Sigma'_{Q,T}$-subformula χ all distinct variables v, v' that occur freely in χ and have column-k-occurrences and column-k'-occurrences in χ, respectively, we know that $k \neq k'$. This observation also entails that any non-trivial equation can be replaced by `false` in Σ'_{QT}-subformulas of ψ_{LGF} and any trivial equation (except for the ones that constitute the guard of a top-most quantification in a Σ_{QT}-sentence) can be replaced by `true`.

Notice that, after the previous modifications, every Σ'_{QT}-subsentence that is part of ψ_{LGF} is actually a variable-renamed version of a loosely guarded FO^n sentence, see Section 3.12.

ψ_S, β_S

Suppose that ψ_{LGF} has fewer than $2^{\uparrow n-1}(n)$ subformulas. We observed earlier that $\mathcal{B} \models \psi_{\mathrm{LGF}}$ and $\mathcal{B}_{-S} \not\models \psi_{\mathrm{LGF}}$ for every $S \in \mathcal{S}_n$. Hence, for every $S \in \mathcal{S}_n$ there is some Σ'_{QT}-subformula ψ_S in ψ_{LGF} of the form $\big(\exists \bar{\mathrm{y}}.\, \gamma_S(\bar{\mathrm{y}}, \bar{\mathrm{z}})\big)\chi_S(\bar{\mathrm{y}}, \bar{\mathrm{z}})$ and some variable assignment β_S such that the following properties hold. We have $\beta_S(y_*) = \mathsf{b}_S^{(n)}$ for exactly one $y_* \in \bar{\mathrm{y}}$ and for every $v \in \bar{\mathrm{y}} \cup \bar{\mathrm{z}}$ different from y_* we have $\beta_S(v) \in \mathsf{A}_{\mathsf{b}} \setminus \mathsf{A}_{\mathsf{b},n}$. Moreover, we have

(∗) $\mathcal{B}, \beta_S \models \gamma_S(\bar{\mathrm{y}}, \bar{\mathrm{z}}) \wedge \chi_S(\bar{\mathrm{y}}, \bar{\mathrm{z}})$ and $\mathcal{B}, \beta' \not\models \gamma_S(\bar{\mathrm{y}}, \bar{\mathrm{z}}) \wedge \chi_S(\bar{\mathrm{y}}, \bar{\mathrm{z}})$ for every β' that differs from β_S only in the value assigned to y_*.

$\bar{\mathsf{c}}_S$

The tuple $\beta_S(\bar{\mathrm{z}})$ represents a sequence $\bar{\mathsf{c}}_S$ of domain elements from A_{b} that can be completed to a chain $\mathsf{b}_{T_1}^{(1)}, \ldots, \mathsf{b}_{T_{n-1}}^{(n-1)}, \mathsf{b}_S^{(n)}$ with $T_1 \in \ldots \in T_{n-1} \in S$.

$\widehat{S}_*$

Fix any $S_* \in \mathcal{S}_n$ and consider the formula $\psi_{S_*}(\bar{\mathrm{z}})$. There is a nonempty set $\widehat{S}_*$ such that $\psi_{S_*}(\bar{\mathrm{z}})$ coincides with every $\psi_S(\bar{\mathrm{z}})$ with $S \in \widehat{S}_*$. For any distinct $S, S' \in \widehat{S}_*$ the sequences $\bar{\mathsf{c}}_S := \beta_S(\bar{\mathrm{z}})$ and $\bar{\mathsf{c}}_{S'} := \beta_{S'}(\bar{\mathrm{z}})$ must differ, for otherwise (∗) would be violated. As there are at most $\prod_{k=1}^{n-1} 2^{\uparrow k}(n)$ distinct sequences $\bar{\mathsf{c}}_S$, $\widehat{S}_*$ can contain at most $\prod_{k=1}^{n-1} 2^{\uparrow k}(n) < \big(2^{\uparrow n-1}(n)\big)^n$ sets. Recall that there are fewer than $2^{\uparrow n-1}(n)$ subformulas in ψ_{LGF}. We have just inferred that each of these can only serve as ψ_S for at most $\big(2^{\uparrow n-1}(n)\big)^n$ sets $S \in \mathcal{S}_n$. Hence, only

$$\big(2^{\uparrow n-1}(n)\big)^n \cdot 2^{\uparrow n-1}(n) \;=\; 2^{(n+1)\cdot 2^{\uparrow n-2}(n)} \;<\; 2^{2^{\uparrow n-1}(n)} \;=\; 2^{\uparrow n}(n)$$

sets S have a corresponding subformula ψ_S. But this means that there are $S \in \mathcal{S}_n$ such that $\mathcal{B}_{-S} \models \psi_{\mathrm{LGF}}$, which contradicts our assumptions. Consequently, ψ_{LGF} must have more than $2^{\uparrow n-1}(n)$ subformulas. □

3.11 Separateness and Guarded Negation

We have already briefly visited the concept of guarded negation in the beginning Chapter 3 (page 26). The used guards $\gamma(\bar{\mathrm{u}})$ are atoms in which every variable from $\bar{\mathrm{u}}$ occurs at least once. An occurrence of negation $\neg\psi(\bar{\mathrm{u}})$ is *guarded*, if it is part of a formula $\gamma(\bar{\mathrm{u}}) \wedge \neg\psi(\bar{\mathrm{u}})$ with an atomic guard $\gamma(\bar{\mathrm{u}})$. The *guarded-negation fragment* comprises first-order sentences in which all quantifiers are existential and every occurrence of negation is guarded by an atomic guard. Under these restrictions, universal quantification can only be expressed in a guarded fashion, simulated by existential quantification. For example, the sentence $\forall \bar{\mathrm{x}}.\, P(\bar{\mathrm{x}}) \to \psi(\bar{\mathrm{x}})$ is equivalent to $\exists y.\, y \approx y \wedge \neg\big(\exists \bar{\mathrm{x}}.\, P(\bar{\mathrm{x}}) \wedge \neg\psi(\bar{\mathrm{x}})\big)$.[5]

Definition 3.11.1 (Guarded-negation fragment (GNFO))**.** *We define the set of* guarded-negation formulas *inductively:*

(i) every relational atom is a guarded-negation formula, equality is admitted;

(ii) every $\wedge$-$\vee$-combination of guarded-negation formulas is a guarded-negation formula;

(iii) for every tuple $\bar{\mathrm{u}}$ and every guarded-negation formula $\psi(\bar{\mathrm{u}})$ the formula $\exists \bar{\mathrm{u}}.\, \psi(\bar{\mathrm{u}})$ is a guarded-negation formula;

[5] This example is an adaptation of an example from [BtCS15], page 3.

(iv) for every tuple $\bar{u}$*, every atomic guard* $\gamma(\bar{u})$ *(i.e.* $\gamma(\bar{u})$ *is an atom in which every* $u \in \bar{u}$ *occurs at least once), and every guarded-negation formula* $\psi(\bar{u})$ *the formula* $\gamma(\bar{u}) \wedge \neg\psi(\bar{u})$ *is a guarded-negation formula.*

The guarded-negation fragment (GNFO) *is the class of all first-order guarded-negation sentences.*

Barany, ten Cate, and Segoufin have shown that GNFO sentences have the same expressive power as GF sentences have (see [BtCS15], Proposition 2.2). Moreover, there are GNFO sentences for which there is no GF equivalent, i.e. GNFO is strictly more expressive than GF. One such example is the sentence $\exists xy.\, E(x,y) \wedge \neg\big(\exists uvw.\, E(x,u) \wedge E(u,v) \wedge E(v,w) \wedge E(w,y)\big)$ ([BtCS15], Example 2.3).

Similar to guarded quantification, guarded negation can be made compatible with separateness of variables in a way that allows us to syntactically extend GNFO while retaining its expressive power and the decidability of the associated satisfiability problem (*GNFO-Sat*). *GNFO-Sat*

Definition 3.11.2 (Separated guarded-negation fragment (SGNFO))**.** *Given any sequence* $\bar{u}_1, \ldots, \bar{u}_n, \bar{v}$ *of pairwise-disjoint tuples of first-order variables, a* separated negation guard $\gamma(\bar{u}_1, \ldots, \bar{u}_n, \bar{v})$ *is a conjunction of* n *atoms* $A_1(\bar{u}_1, \bar{v}) \wedge \ldots \wedge A_n(\bar{u}_n, \bar{v})$ *(possibly equations) such that for every* i*,* $1 \leq i \leq n$*, all variables from* $\bar{u}_i$ *occur at least once in* $A_i(\bar{u}_i, \bar{v})$*.*

We define the set of separated guarded-negation formulas *inductively:*

(i) every relational atom is a separated guarded-negation formula, equality is admitted;

(ii) every $\wedge$*-*$\vee$*-combination of separated guarded-negation formulas is a separated guarded-negation formula;*

(iii) for every tuple $\bar{y}$ *and every separated guarded-negation formula* $\psi(\bar{y})$ *the formula* $\exists\bar{y}.\, \psi(\bar{y})$ *is a separated guarded-negation formula;*

(iv) for every separated negation guard $\gamma(\bar{u}_1, \ldots, \bar{u}_n, \bar{v})$*, and every separated guarded-negation formula* $\psi(\bar{u}_1, \ldots, \bar{u}_n)$ *the formula* $\gamma(\bar{u}_1, \ldots, \bar{u}_n, \bar{v}) \wedge \neg\psi(\bar{u}_1, \ldots, \bar{u}_n)$ *is a separated guarded-negation formula if the following conditions are met. Let* Z *be the set of variables that are quantified in* $\psi(\bar{u}_1, \ldots, \bar{u}_n)$*. We require that* Z *can be divided into pairwise disjoint, possibly empty subsets* $Z_1, \ldots, Z_n$ *such that the sets* $Z_1 \cup \bar{u}_1, \ldots, Z_n \cup \bar{u}_n$ *are all pairwise separated in* $\psi(\bar{u}_1, \ldots, \bar{u}_n)$*.*

The separated guarded-negation fragment (SGNFO) *is the set of all first-order separated guarded-negation sentences.*

We shall occasionally use sloppy language and speak of GNFO and SGNFO *formulas* when we mean (separated) negation-guarded formulas that are not necessarily closed.

It is obvious that GNFO is contained in SGNFO and that there are SGNFO sentences that do not belong to GNFO. Moreover, every MFO sentence φ can be easily turned into an equivalent SGNFO sentence with a length linear in the original. We first transform φ into negation normal form and add trivial equations $v \approx v$ as guards to negated atomic subformulas $\neg P(v)$. The result lies in the intersection of SGNFO and $\mathrm{MFO}_{\approx}$. For $\mathrm{MFO}_{\approx}$ sentences the matter seems to be more complicated. The sentence $\exists xy.\, x \not\approx y$, for instance, is not an SGNFO sentence and does not seem to have an SGNFO equivalent.

Proposition 3.11.3. *SGNFO properly contains GNFO. Moreover, every MFO sentence can be turned into an equivalent SGNFO sentence with a formula length that is linear in the length of the original.*

After we have seen the results obtained for the other novel first-order fragments, it should not come as a surprise that there is an effective translation from SGNFO to GNFO.

Lemma 3.11.4. *Every SGNFO formula is equivalent to some GNFO formula.*

Proof. First, we recall the notion of *strict separateness* from Lemma 2.0.3. Let ψ be any first-order formula. Two disjoint sets of first-order variables X, Y are *strictly separated in* ψ if X and Y are separated in ψ and, in addition, for every subformula $\psi' := (\mathcal{Q}v.\ \ldots)$ of ψ we either have $\mathrm{vars}(\psi') \cap X = \emptyset$ or $\mathrm{vars}(\psi') \cap Y = \emptyset$.

We infer with two auxiliary results from which the lemma follows:

Claim I: Consider any SGNFO formula $\varphi(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n, \bar{\mathrm{v}}) := \gamma(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n, \bar{\mathrm{v}}) \wedge \neg\psi(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$ where $\psi(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$ is any GNFO formula, and the $\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n$ are pairwise strictly separated in $\psi(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$. Then, $\varphi(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n, \bar{\mathrm{v}})$ is equivalent to some GNFO formula $\varphi'(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n, \bar{\mathrm{v}})$ in which $\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n$ are pairwise strictly separated.

basic formulas

Proof: Let *basic formulas* in $\psi(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$ be subformulas that do not lie in the scope of any quantifier or negation sign in $\psi(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$ and that are either guarded negation formulas $\delta(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_k, \bar{\mathrm{y}}) \wedge \neg\chi(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_k)$, quantified formulas $\exists\bar{\mathrm{y}}.\, \chi(\bar{\mathrm{y}}, \bar{\mathrm{x}})$, or atoms. Transform $\psi(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$ into a conjunction $\psi' := \bigwedge_{i\in I} \eta_i(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$ of disjunctions $\eta_i(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$ of basic formulas. Since we assumed $\psi(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$ to be a GNFO formula and that the $\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n$ are pairwise strictly separated in $\psi(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$, we conclude that every basic formula $\chi(\bar{\mathrm{x}})$ in $\psi(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$ satisfies $\bar{\mathrm{x}} \cap \bar{\mathrm{u}}_\ell \neq \emptyset$ for at most one ℓ, $1 \leq \ell \leq n$. Hence, the disjuncts $\eta_i(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$ in $\psi'(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$ can be regrouped such that

$$\psi' = \bigwedge_{i\in I} \eta_{i,1}(\bar{\mathrm{u}}_1) \vee \ldots \vee \eta_{i,n}(\bar{\mathrm{u}}_n)\ .$$

Therefore, $\gamma(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n, \bar{\mathrm{v}}) \wedge \neg\psi(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$ is equivalent to the following sentence, where $A_1(\bar{\mathrm{u}}_1, \bar{\mathrm{v}}), \ldots, A_n(\bar{\mathrm{u}}_n, \bar{\mathrm{v}})$ is the list of atoms that $\gamma(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n, \bar{\mathrm{v}})$ comprises:

$$\begin{aligned}
&A_1(\bar{\mathrm{u}}_1, \bar{\mathrm{v}}) \wedge \ldots \wedge A_n(\bar{\mathrm{u}}_n, \bar{\mathrm{v}}) \wedge \neg\psi'(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)\\
\models\ &A_1(\bar{\mathrm{u}}_1, \bar{\mathrm{v}}) \wedge \ldots \wedge A_n(\bar{\mathrm{u}}_n, \bar{\mathrm{v}}) \wedge \neg \bigwedge_{i\in I} \eta_{i,1}(\bar{\mathrm{u}}_1) \vee \ldots \vee \eta_{i,n}(\bar{\mathrm{u}}_n)\\
\models\ &A_1(\bar{\mathrm{u}}_1, \bar{\mathrm{v}}) \wedge \ldots \wedge A_n(\bar{\mathrm{u}}_n, \bar{\mathrm{v}}) \wedge \bigvee_{i\in I} \neg\eta_{i,1}(\bar{\mathrm{u}}_1) \wedge \ldots \wedge \neg\eta_{i,n}(\bar{\mathrm{u}}_n)\\
\models\ &\bigvee_{i\in I} \big(A_1(\bar{\mathrm{u}}_1, \bar{\mathrm{v}}) \wedge \neg\eta_{i,1}(\bar{\mathrm{u}}_1)\big) \wedge \ldots \wedge \big(A_n(\bar{\mathrm{u}}_n, \bar{\mathrm{v}}) \wedge \neg\eta_{i,n}(\bar{\mathrm{u}}_n)\big)
\end{aligned}$$

This is the sought GNFO formula. ◊

Claim II: Consider any SGNFO formula $\varphi(\bar{\mathrm{x}}, \bar{\mathrm{v}}) := \exists\bar{\mathrm{y}}.\, \psi(\bar{\mathrm{y}}, \bar{\mathrm{x}}, \bar{\mathrm{v}})$ where $\psi(\bar{\mathrm{y}}, \bar{\mathrm{x}}, \bar{\mathrm{v}})$ is any GNFO formula in which the sets $\bar{\mathrm{y}} \cup \bar{\mathrm{x}}$ and $\bar{\mathrm{v}}$ are strictly separated. Then, $\varphi(\bar{\mathrm{x}}, \bar{\mathrm{v}})$ is equivalent to some GNFO formula $\varphi(\bar{\mathrm{x}}, \bar{\mathrm{v}})$ in which $\bar{\mathrm{y}} \cup \bar{\mathrm{x}}$ and $\bar{\mathrm{v}}$ are strictly separated.

basic formulas

Proof: Let *basic formulas* in $\psi(\bar{\mathrm{y}}, \bar{\mathrm{x}}, \bar{\mathrm{v}})$ be defined like in the proof of Claim I. Transform $\psi(\bar{\mathrm{y}}, \bar{\mathrm{x}}, \bar{\mathrm{v}})$ into a disjunction $\psi' := \bigvee_{i\in I} \eta_i(\bar{\mathrm{y}}, \bar{\mathrm{x}}, \bar{\mathrm{v}})$ of conjunctions $\eta_i(\bar{\mathrm{y}}, \bar{\mathrm{x}}, \bar{\mathrm{v}})$ of basic formulas. Since we assumed $\psi(\bar{\mathrm{y}}, \bar{\mathrm{x}}, \bar{\mathrm{v}})$ to be a GNFO formula and that the sets $\bar{\mathrm{y}} \cup \bar{\mathrm{x}}$ and $\bar{\mathrm{v}}$ are strictly separated in $\psi(\bar{\mathrm{y}}, \bar{\mathrm{x}}, \bar{\mathrm{v}})$, every basic formula $\chi(\bar{\mathrm{u}})$ in $\psi(\bar{\mathrm{y}}, \bar{\mathrm{x}}, \bar{\mathrm{v}})$ satisfies $\bar{\mathrm{u}} \cap (\bar{\mathrm{y}} \cup \bar{\mathrm{x}}) = \emptyset$ or $\bar{\mathrm{u}} \cap \bar{\mathrm{v}} = \emptyset$. Hence, the conjuncts $\eta_i(\bar{\mathrm{y}}, \bar{\mathrm{x}}, \bar{\mathrm{v}})$ in $\psi'(\bar{\mathrm{y}}, \bar{\mathrm{x}}, \bar{\mathrm{v}})$ can be regrouped such that

$$\psi' = \bigvee_{i\in I} \eta_{i,1}(\bar{\mathrm{y}}, \bar{\mathrm{x}}) \wedge \eta_{i,2}(\bar{\mathrm{v}})\ .$$

Therefore, $\exists\bar{\mathrm{y}}.\, \psi(\bar{\mathrm{y}}, \bar{\mathrm{x}}, \bar{\mathrm{v}})$ is equivalent to the sentence

$$\begin{aligned}
&\exists\bar{\mathrm{y}}.\, \bigvee_{i\in I} \eta_{i,1}(\bar{\mathrm{y}}, \bar{\mathrm{x}}) \vee \eta_{i,2}(\bar{\mathrm{v}})\\
\models\ &\bigvee_{i\in I} \big(\exists\bar{\mathrm{y}}.\, \eta_{i,1}(\bar{\mathrm{y}}, \bar{\mathrm{x}})\big) \wedge \eta_{i,2}(\bar{\mathrm{v}})\ .
\end{aligned}$$

This is the sought GNFO formula. ◊

Now consider any SGNFO formula φ that is not a GNFO formula. Let $\chi(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n, \bar{\mathrm{v}}) := \gamma(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n, \bar{\mathrm{v}}) \wedge \neg\chi'(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$ be a smallest subformula of φ that violates the conditions of guarded negation in GNFO. Hence, the set Z of variables quantified in $\chi'(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$ can be divided into pairwise disjoint sets $Z_1, \ldots, Z_n$ such that $Z_1 \cup \bar{\mathrm{u}}_1, \ldots, Z_n \cup \bar{\mathrm{u}}_n$ are pairwise separated in $\chi'(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$. Further suppose that in $\chi'(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$ the sets $Z_1 \cup \bar{\mathrm{u}}_1, \ldots, Z_n \cup \bar{\mathrm{u}}_n$ are not strictly separated. Let $\eta(\bar{\mathrm{x}}) := \exists\bar{\mathrm{y}}.\, \eta'(\bar{\mathrm{y}}, \bar{\mathrm{x}})$ be a smallest subformula of $\chi'(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$ that violates this strict-separateness condition. Hence, we can subdivide $\bar{\mathrm{y}}$ into pairwise disjoint parts $\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n$ such that $\bar{\mathrm{y}}_i \subseteq Z_i$ for every i. Moreover, we can subdivide $\bar{\mathrm{x}}$ into pairwise disjoint parts $\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n$ such that $\bar{\mathrm{x}}_i \subseteq Z_i \cup \bar{\mathrm{u}}_i$ for every i. Then, $\eta(\bar{\mathrm{x}})$ can be rewritten into $\exists\bar{\mathrm{y}}_1\exists\bar{\mathrm{y}}_2 \ldots \exists\bar{\mathrm{y}}_n.\, \eta'(\bar{\mathrm{y}}_1, \bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{y}}_n, \bar{\mathrm{x}}_n)$. Since we assume $\eta(\bar{\mathrm{x}})$ to be minimal, the sets $\bar{\mathrm{y}}_1 \cup \bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{y}}_n \cup \bar{\mathrm{x}}_n$ are pairwise strictly separated in $\eta'(\bar{\mathrm{y}}_1, \bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{y}}_n, \bar{\mathrm{x}}_n)$. By Claim II, $\eta(\bar{\mathrm{x}})$ is equivalent to some $\eta''(\bar{\mathrm{x}})$ in which the $Z_1 \cup \bar{\mathrm{u}}_1, \ldots, Z_n \cup \bar{\mathrm{u}}_n$ are pairwise strictly separated. Therefore, the formula $\chi'(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$ can be transformed into an equivalent formula $\chi''(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$ in which $Z_1 \cup \bar{\mathrm{u}}_1, \ldots, Z_n \cup \bar{\mathrm{u}}_n$ are pairwise strictly separated. By Claim I, $\gamma(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n, \bar{\mathrm{v}}) \wedge \neg\chi''(\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n)$ can be transformed into an equivalent formula that belongs to GNFO and in which the sets $\bar{\mathrm{u}}_1, \ldots, \bar{\mathrm{u}}_n$ are pairwise strictly separated.

By iterative and exhaustive application of the outlined transformation, we can derive a GNFO formula that is equivalent to the SGNFO formula φ. □

Since GNFO is known to possess the finite model property [BtCS15], Lemma 3.11.4 entails the same for SGNFO. Of course, this also means that the satisfiability problem associated with SGNFO (*SGNFO-Sat*) is decidable. *SGNFO-Sat*

Theorem 3.11.5. *SGNFO possess the finite model property and, hence, the satisfiability problem for SGNFO sentences is decidable.*

3.12 Separateness and Finite-Variable First-Order Logic

The class of first-order formulas over a fixed finite set of variables yields an interesting object of study (see, e.g., [Ott97, Gro98, Daw99, GO99, KPHT18] and also the textbooks [Lib04], Section 11, and [GKL$^+$07], Sections 1.1.3, 2.7, and 2.8). The special case where only two variables are admitted gives rise to the *two-variable fragment* of first-order logic, FO^2, that we have already briefly discussed on page 25. It is important to understand that this restriction allows reusing variable names in nested quantifiers. Therefore, in the formulas in the present section we explicitly allow variables to occur free and bound in a formula, and to reappear in distinct occurrences of quantifiers in the same formula. For example, the sentence $\forall x\exists y.\, \big(E(x,y) \wedge \exists x.\, \big(E(y,x) \wedge \exists y.\, E(x,y)\big)\big)$ belongs to FO^2. It stipulates the existence of a path of length at least three, starting from any node in a directed graph.

We shall see in this section that also in the context of finite-variable logics separateness can give us more syntactic freedom and the ability to express certain properties in a substantially more succinct way, on the one hand. On the other hand, the overall expressive power is retained, if restrictions are formulated in the right way.

Definition 3.12.1 (Separated finite-variable formulas)**.** *For any positive integer k we define* FO^k *to be the set of all relational first-order formulas in which all variables are taken from a finite sequence $x_1, \ldots, x_k$.* FO^k

For every $k \geq 1$ we define the class SFO^k *of relational first-order formulas as follows. Let $V_1, V_2, V_3, \ldots$ be a sequence of pairwise disjoint sets $V_i \subseteq$ Var of first-order variables, each containing exactly k pairwise distinct variables. For every $m \geq 1$ we define the set* SFO^{k,m} *to be the set of all relational first-order formulas φ in which all variables are taken from $V_1 \cup \ldots \cup V_m$ and in which all sets $V_1, \ldots, V_m$ are pairwise separated. The class* SFO^k *is the union* $\bigcup_{m\geq 1}$ SFO^{k,m}.

It is easy to see that FO^k is a special case of SFO^k. Moreover, MFO is a proper subset of SFO^k for $k = 1$. In contrast, for every positive integer k the MFO$_\approx$ sentence $\forall x_1 \ldots x_k\exists y.\, \bigwedge_k y \not\approx x_k$ does not belong to SFO^k.

Proposition 3.12.2. *For every positive k,* SFO^k *contains* FO^k *and MFO.*

In the following lemma we establish the equivalence between SFO^k and $\mathrm{FO}^k$for every positive k by devising an equivalence-preserving translation procedure between the two sets.

Lemma 3.12.3. *Every* SFO^k *sentence is equivalent to some* FO^k *sentence.*

Proof. Let m be any positive integer and consider any sentence φ from $\mathrm{SFO}^{k,m}$. Then, $\mathrm{vars}(\varphi) \subseteq V_1 \cup \ldots \cup V_m$ and all $V_1, \ldots, V_m$ are pairwise separated in φ. Without loss of generality, we assume that φ is in negation normal form.

We prove an auxiliary result from which the lemma follows.

Claim I: Consider any subformula $\psi = \mathcal{Q}\bar{\mathrm{v}}.\,\chi$ of φ with $\bar{\mathrm{v}} \subseteq V_i$ for some i. If the sets $V_1, \ldots, V_m$ are pairwise strictly separated in χ, then we can construct a formula ψ' that is equivalent to ψ and in which all sets $V_1, \ldots, V_m$ are pairwise strictly separated.

Proof: The proof proceeds along the same lines as the proof of Lemma 2.0.3.

basic formulas

A *basic formula* is any atom and any subformula $(\mathcal{Q}v.\ldots)$ in χ that does not lie within the scope of any quantifier in χ. Suppose $\mathcal{Q}$ is an existential quantifier. (The case of $\mathcal{Q} = \forall$ can be treated in an analogous way.)

Let $\bar{\mathrm{z}}$ be the tuple collecting all variables that occur freely in ψ. We first transform χ into an equivalent disjunction of conjunctions of negated or non-negated basic formulas. This is always possible. Since the sets $V_1, \ldots, V_m$ are pairwise strictly separated in χ, the constituents of the j-th conjunction can be grouped into m parts: $\eta_{j,1}\big(V_1 \cap (\bar{\mathrm{v}} \cup \bar{\mathrm{z}})\big), \ldots, \eta_{j,m}\big(V_m \cap (\bar{\mathrm{v}} \cup \bar{\mathrm{z}})\big)$ with $\mathrm{vars}(\eta_{j,\ell}) \subseteq V_\ell$. This is possible because of our assumption that the sets $V_1, \ldots, V_m$ are all pairwise strictly separated in χ. Hence, since $\bar{\mathrm{v}} \subseteq V_i$, ψ is equivalent to a formula of the form

$$\exists\bar{\mathrm{v}}.\bigvee_j \eta_{j,i}\big(V_i \cap (\bar{\mathrm{v}} \cup \bar{\mathrm{z}})\big) \wedge \bigwedge_{\substack{1 \le \ell \le m \\ \ell \neq i}} \eta_{j,\ell}(V_\ell \cap \bar{\mathrm{z}}) \ .$$

We shift the existential quantifier block $\exists\bar{\mathrm{v}}$ inwards so that it only binds the (sub-)conjunctions $\eta_{j,i}\big(V_i \cap (\bar{\mathrm{v}} \cup \bar{\mathrm{z}})\big)$. The resulting formula

$$\bigvee_j \Big(\exists\bar{\mathrm{v}}.\,\eta_{j,i}\big(V_i \cap (\bar{\mathrm{v}} \cup \bar{\mathrm{z}})\big)\Big) \wedge \bigwedge_{\substack{1 \le \ell \le m \\ \ell \neq i}} \eta_{j,\ell}(V_\ell \cap \bar{\mathrm{z}})$$

is the sought ψ' in which the sets $V_1, \ldots, V_m$ are all pairwise strictly separated. ◊

Clearly, the sets $V_1, \ldots, V_m$ are pairwise strictly separated in any quantifier-free subformula of φ. Hence, applying Claim I iteratively, we can transform φ into an equivalent sentence φ' in which the sets $V_1, \ldots, V_m$ are pairwise strictly separated. Since φ' is a sentence, the strict separateness condition leads to the observation that for every subformula $\mathcal{Q}\bar{\mathrm{v}}.\,\chi$ in φ' there is some j such that $\mathrm{vars}(\mathcal{Q}\bar{\mathrm{v}}.\,\chi) \subseteq V_j$. As each of the V_i contains exactly k variables, we can rename the bound variables in φ' such that φ' is an $\wedge$-$\vee$-combination of FO^k sentences. Since FO^k is closed under Boolean combinations, φ' is an FO^k sentence. □

Since the satisfiability problem for FO^2 sentences is known to be decidable — in fact, the class of FO^2 sentences is known to possess the finite model property [Mor75, GKV97] —, Lemma 3.12.3 entails the same for the class of SFO^2 sentences and the associated satisfiability problem SFO^2-*Sat*.

SFO^2-*Sat*

Theorem 3.12.4. *The class of* SFO^2 *sentences possess the finite model property and, hence, the satisfiability problem for* SFO^2 *sentences is decidable.*

Having established the equivalence between FO^k and SFO^k regarding expressiveness, it remains to investigate the succinctness gap between the two fragments. We shall do this in particular for the class of SFO^2 sentences compared to the class of FO^2 sentences.

Theorem 3.12.5. *There is a class of* SFO^2 *sentences and some positive integer* n_0 *such that for every integer* $n \geq n_0$ *the class contains a sentence* φ *with a length linear in* n *for which any equivalent* FO^2 *sentence has a length that is at least exponential in* n.

Proof. Let $n \geq 1$ be some positive integer. Consider the following first-order sentence in which the sets $\{x_1, x_2\}$ and $\{y_1, y_2\}$ are separated:

$$\varphi := \forall x_2 \exists y_2 \forall x_1 \exists y_1 . \bigwedge_{i=1}^{2(n+1)} \big(P_i(x_1, x_2) \leftrightarrow Q_i(y_1, y_2)\big) \ .$$

In analogy to the proof of Theorem 3.2.7, we construct the following model $\mathcal{A}$ for φ. The construction is based on the sets $\mathcal{S}_1 := \big\{S \subseteq [2(n+1)] \;\big|\; |S| = n+1\big\}$ and $\mathcal{S}_2 := \big\{S \subseteq \mathcal{S}_1 \;\big|\; |S| = \frac{1}{2}|\mathcal{S}_1|\big\}$. We observe $\mathcal{S}_k$

$$|\mathcal{S}_1| = \binom{2(n+1)}{n+1} \;\geq\; \left(\frac{2(n+1)}{n+1}\right)^{n+1} \;=\; 2^{n+1}$$

and

$$|\mathcal{S}_2| = \binom{|\mathcal{S}_1|}{|\mathcal{S}_1|/2} \;\geq\; \left(\frac{|\mathcal{S}_1|}{|\mathcal{S}_1|/2}\right)^{|\mathcal{S}_1|/2} \;\geq\; 2^{2^n} \ ,$$

in analogy to the proof of Theorem 3.2.7.

Claim I: Let $\widehat{\mathcal{S}}$ be any subset of $\mathcal{S}_2$ such that for every $S \in \widehat{\mathcal{S}}$ there is some $T \in S \subseteq \mathcal{S}_1$ which does not belong to any $S' \in \widehat{\mathcal{S}} \setminus \{S\}$. Then, $\widehat{\mathcal{S}}$ contains at most $|\mathcal{S}_1| \leq 2^{2(n+1)}$ sets as elements.

Proof: Obvious. ◊

Let $\mathcal{A}$ be the structure with $\mathcal{A}$

$$\mathsf{A} := \big\{\mathsf{a}_S^{(1)}, \mathsf{b}_S^{(1)} \;\big|\; S \in \mathcal{S}_1\big\} \cup \big\{\mathsf{a}_S^{(2)}, \mathsf{b}_S^{(2)} \;\big|\; S \in \mathcal{S}_2\big\},$$

$$P_i^{\mathcal{A}} := \big\{\langle \mathsf{a}_{S_1}^{(1)}, \mathsf{a}_{S_2}^{(2)}\rangle \in \mathsf{A} \times \mathsf{A} \;\big|\; i \in S_1 \in S_2\big\} \text{ for } i = 1, \ldots, 2(n+1), \text{ and}$$

$$Q_i^{\mathcal{A}} := \big\{\langle \mathsf{b}_{S_1}^{(1)}, \mathsf{b}_{S_2}^{(2)}\rangle \in \mathsf{A} \times \mathsf{A} \;\big|\; i \in S_1 \in S_2\big\} \text{ for } i = 1, \ldots, 2(n+1).$$

Then, for any choice of S_1, S_2 and every i, $1 \leq i \leq 2(n+1)$, we have

$$\mathcal{A}, \big[x_1 \mapsto \mathsf{a}_{S_1}^{(1)}, x_2 \mapsto \mathsf{a}_{S_2}^{(2)}, y_1 \mapsto \mathsf{b}_{S_1}^{(1)}, y_2 \mapsto \mathsf{b}_{S_2}^{(2)}\big] \models P_i(x_1, x_2) \leftrightarrow Q_i(y_1, y_2) \ .$$

For any other choice of pairs $\langle \mathsf{c}_1, \mathsf{c}_2\rangle$, i.e. there do not exist sets $S_1 \in \mathcal{S}_1, S_2 \in \mathcal{S}_2$ such that $\langle \mathsf{c}_1, \mathsf{c}_2\rangle$ equals $\langle \mathsf{a}_{S_1}^{(1)}, \mathsf{a}_{S_2}^{(2)}\rangle$ or $\langle \mathsf{b}_{S_1}^{(1)}, \mathsf{b}_{S_2}^{(2)}\rangle$, we observe $\mathcal{A}, [x_1 \mapsto \mathsf{c}_1, x_2 \mapsto \mathsf{c}_2] \not\models P_i(x_1, x_2)$ and $\mathcal{A}, [y_1 \mapsto \mathsf{c}_1, y_2 \mapsto \mathsf{c}_2] \not\models Q_i(y_1, y_2)$ for every i. Hence,

$$\mathcal{A}, [x_1 \mapsto \mathsf{c}_1, x_2 \mapsto \mathsf{c}_2, y_1 \mapsto \mathsf{c}_1, y_2 \mapsto \mathsf{c}_2] \models \bigwedge_{i=1}^{2(n+1)} P_i(x_1, x_2) \leftrightarrow Q_i(y_1, y_2) \ .$$

Consequently, $\mathcal{A}$ is a model of φ.

In analogy to the proof of Theorem 3.2.7, we can prove the following observation.

Claim II: For every $S \in \mathcal{S}_2$ the substructure $\mathcal{A}_{-S}$ of $\mathcal{A}$ induced by $\mathsf{A}_{-S} := \mathsf{A} \setminus \big\{\mathsf{b}_S^{(2)}\big\}$ does not satisfy φ. ◊ $\mathcal{A}_{-S}$

Let φ_{FO^2} be a shortest FO^2 sentence that is semantically equivalent to φ. Next, we argue that $\mathrm{len}(\varphi_{\mathrm{FO}^2})$ is at least exponential in n. In [Sco62] a normal form for FO^2 sentences was introduced, which is sometimes referred to as *Scott normal form* in the literature, e.g. in [GO99]. Accordingly, Lemma 8.1.2 in [BGG97] states that there is some relational FO^2 sentence ψ_{FO^2} that has the following properties: *Scott normal form*

(a) ψ_{FO^2} is of the form $\big(\forall uv.\,\chi(u,v)\big) \wedge \bigwedge_{i=1}^{m} \forall x \exists y.\,\eta_i(x,y)$ with quantifier-free χ and η_i,

(b) the vocabulary underlying ψ_{FO^2} is that of φ_{FO^2} extended by fresh unary predicate symbols $R_1, \ldots, R_\kappa$ with $\kappa \in \mathcal{O}\big(\mathrm{len}(\varphi_{\mathrm{FO}^2})\big)$, — κ

(c) $\psi_{\mathrm{FO}^2} \models \varphi_{\mathrm{FO}^2}$,

(d) every model of φ_{FO^2} can be uniquely expanded to a model of ψ_{FO^2} over the same domain and conserving the interpretations of all predicate symbols occurring in φ_{FO^2}, and

(e) $\mathrm{len}(\psi_{\mathrm{FO}^2}) \in \mathcal{O}\big(\mathrm{len}(\varphi_{\mathrm{FO}^2})\big)$.

$\mathcal{B}$ — Let $\mathcal{B}$ be the unique expansion of $\mathcal{A}$ for which $\mathcal{B} \models \psi_{\mathrm{FO}^2}$ and $\mathsf{B} := \mathsf{A}$. Claim II can be extended to $\mathcal{B}$, because of $\psi_{\mathrm{FO}^2} \models \varphi_{\mathrm{FO}^2}$. The set $\{\mathsf{b}_S^{(2)} \mid S \in \mathcal{S}_2\}$ can be partitioned into at most 2^κ parts, $\widehat{\mathsf{D}}$ — each containing elements that are indistinguishable by their belonging to the sets $R_k^{\mathcal{B}}$. Let $\widehat{\mathsf{D}}$ be the largest of these parts and let $\widehat{\mathcal{S}} := \{S \mid \mathsf{b}_S^{(2)} \in \widehat{\mathsf{D}}\}$. Hence, for all $\mathsf{b}, \mathsf{b}' \in \widehat{\mathsf{D}}$ and every k with $1 \leq k \leq \kappa$ we have $\mathsf{b} \in R_k^{\mathcal{B}}$ if and only if $\mathsf{b}' \in R_k^{\mathcal{B}}$.

Claim III: Let n be sufficiently large. If κ is polynomial in n, then there is some $S_* \in \widehat{\mathcal{S}}$ such that $\mathsf{b}_{S_*}^{(2)} \in \widehat{\mathsf{D}}$ and for every $T \in S_*$ there is some $S' \in \widehat{\mathcal{S}} \setminus \{S_*\}$ that also contains T and we have $\mathsf{b}_{S'}^{(2)} \in \widehat{\mathsf{D}}$.

Proof: Clearly, $\widehat{\mathsf{D}}$ contains at least $2^{2^n}/2^\kappa = 2^{2^n-\kappa}$ domain elements. Hence, $|\widehat{\mathcal{S}}| \geq 2^{2^n-\kappa}$. Moreover, we observe $2^{2^n-\kappa} > 2^{2(n+1)}$ for sufficiently large n, if κ is polynomial in n. By Claim I, there is some $S_* \in \widehat{\mathcal{S}}$ such that for every $T \in S_*$ there is some $S' \in \widehat{\mathcal{S}} \setminus \{S_*\}$ with $T \in S'$. Claim III follows by definition of $\widehat{\mathsf{D}}$ and $\widehat{\mathcal{S}}$. ◊

S_*, $\mathcal{B}_*$, J — We fix some $S_* \in \widehat{\mathcal{S}}$ as described in Claim III. Let $\mathcal{B}_{-S_*}$ be the substructure of $\mathcal{B}$ induced by the domain $\mathsf{B}_* := \mathsf{B} \setminus \{\mathsf{b}_{S_*}^{(2)}\}$. By Claim II (extended to $\mathcal{B}$), there is some maximal nonempty set $J \subseteq [m]$ such that for every $j \in J$ we have $\mathcal{B}_{-S_*} \not\models \forall x \exists y.\,\eta_j(x,y)$. Consequently, for every $j \in J$ there is some domain element $\mathsf{c} \in \mathsf{B}_*$ such that $\mathcal{B}, [x{\mapsto}\mathsf{c}, y{\mapsto}\mathsf{b}_{S_*}^{(2)}] \models \eta_j(x,y)$ and $\mathcal{B}, [x{\mapsto}\mathsf{c}, y{\mapsto}\mathsf{d}] \not\models \eta_j(x,y)$ for every $\mathsf{d} \in \mathsf{B} \setminus \{\mathsf{b}_{S_*}^{(2)}\}$. Regarding the domain element c, we distinguish two cases.

Consider any $j \in J$ and any $\mathsf{c} \in \mathsf{B}_* \setminus \{\mathsf{b}_S^{(1)} \mid S \in \mathcal{S}_1\}$ for which we have $\mathcal{B}, [x{\mapsto}\mathsf{c}, y{\mapsto}\mathsf{d}] \not\models \eta_j(x,y)$ for every $\mathsf{d} \in \mathsf{B}_*$. Let S' be some set from $\widehat{\mathcal{S}}$ that is different from S_* and for which $\mathsf{c} \neq \mathsf{b}_{S'}^{(2)}$. Notice that η_j is quantifier free and, hence, exclusively contains atoms over the variables x, y. Moreover, for every binary atom A of the form $P_i(x,y)$, $P_i(y,x)$, $Q_i(x,y)$, or $Q_i(y,x)$ we have $\mathcal{B}, [x{\mapsto}\mathsf{c}, y{\mapsto}\mathsf{d}] \not\models A$ for every $\mathsf{d} \in \{\mathsf{b}_S^{(2)} \mid S \in \mathcal{S}_2\}$, including $\mathsf{b}_{S_*}^{(2)}$ and $\mathsf{b}_{S'}^{(2)}$. Since all other non-equational atoms occurring in η_j are monadic and because of $\mathsf{b}_{S_*}^{(2)}, \mathsf{b}_{S'}^{(2)} \in \widehat{\mathsf{D}}$, we conclude the following. For every non-equational atom A occurring in η_j we have $\mathcal{B}, [x{\mapsto}\mathsf{c}, y{\mapsto}\mathsf{b}_{S_*}^{(2)}] \models A(x,y)$ if and only if $\mathcal{B}, [x{\mapsto}\mathsf{c}, y{\mapsto}\mathsf{b}_{S'}^{(2)}] \models A(x,y)$. Consider any equation $x \approx y$. Because of $\mathsf{c} \in \mathsf{B}_{-S_*}$, we have $\mathcal{B} \not\models \mathsf{c} \approx \mathsf{b}_{S_*}^{(2)}$. On the other hand, we also have $\mathcal{B}_{-S_*} \not\models \mathsf{c} \approx \mathsf{b}_{S'}^{(2)}$. But then, we all in all get $\mathcal{B}_{-S_*}, [x{\mapsto}\mathsf{c}, y{\mapsto}\mathsf{b}_{S'}^{(2)}] \models \eta_j(x,y)$, which entails $\mathcal{B}_{-S_*}, [x{\mapsto}\mathsf{c}] \models \exists y.\,\eta_j(x,y)$. This leads to a contradiction and there, hence, cannot be a pair j, c as described.

Consider any $j \in J$ and any $\mathsf{c} \in \{\mathsf{b}_S^{(1)} \mid S \in \mathcal{S}_1\}$ for which $\mathcal{B}, [x{\mapsto}\mathsf{c}, y{\mapsto}\mathsf{d}] \not\models \eta_j(x,y)$ for every $\mathsf{d} \in \mathsf{B}_*$. Hence, there is some set $T \in \mathcal{S}_1$ such that $\mathsf{c} = \mathsf{b}_T^{(1)}$. Suppose $T \notin S_*$. Then, for every binary atom A of the form $P_i(x,y)$, $P_i(y,x)$, $Q_i(x,y)$, $Q_i(y,x)$, or $x \approx y$ we have $\mathcal{B}, [x{\mapsto}\mathsf{c}, y{\mapsto}\mathsf{d}] \not\models A$ for every $\mathsf{d} \in \{\mathsf{b}_S^{(2)} \mid S \in \mathcal{S}_2\}$, including $\mathsf{b}_{S_*}^{(2)}$ and any $\mathsf{b}_{S'}^{(2)} \in \widehat{\mathsf{D}} \setminus \{\mathsf{b}_{S_*}^{(2)}\}$. Like in the above case we conclude $\mathcal{B}_{-S_*}, [x{\mapsto}\mathsf{c}] \models \exists y.\,\eta_j(x,y)$, which yields a contradiction. Suppose $T \in S_*$. By Claim III, there is some $S' \in \widehat{\mathcal{S}} \setminus \{S_*\}$ such that $T \in S'$ and $\mathsf{b}_{S'}^{(2)} \in \widehat{\mathsf{D}} \setminus \{\mathsf{b}_{S_*}^{(2)}\} \subseteq \mathsf{B}_*$. Then, we have

$$\mathcal{B}, [x{\mapsto}\mathsf{b}_T^{(1)}, y{\mapsto}\mathsf{b}_{S_*}^{(2)}] \models Q_i(x,y) \quad \text{if and only if } i \in T$$

c, T

and

$$\mathcal{B}, [x{\mapsto}\mathfrak{b}^{(1)}_T, y{\mapsto}\mathfrak{b}^{(2)}_{S'}] \models Q_i(x,y) \quad \text{ if and only if } i \in T \ .$$

For every other binary atom A of the form $Q_i(y,x)$, $P_i(x,y)$, $P_i(y,x)$, or $x \approx y$ we have $\mathcal{B}, [x{\mapsto}\mathfrak{b}^{(1)}_T, y{\mapsto}\mathfrak{b}^{(2)}_{S_*}] \not\models A(x,y)$ and $\mathcal{B}, [x{\mapsto}\mathfrak{b}^{(1)}_T, y{\mapsto}\mathfrak{b}^{(2)}_{S'}] \not\models A(x,y)$. For every monadic atom A occurring in η_j we have

$$\mathcal{B}, [x{\mapsto}\mathfrak{b}^{(1)}_T, y{\mapsto}\mathfrak{b}^{(2)}_{S_*}] \models A \quad \text{ if and only if } \quad \mathcal{B}, [x{\mapsto}\mathfrak{b}^{(1)}_T, y{\mapsto}\mathfrak{b}^{(2)}_{S'}] \models A \ .$$

All in all, this leads to

$$\mathcal{B}, [x{\mapsto}\mathfrak{b}^{(1)}_T, y{\mapsto}\mathfrak{b}^{(2)}_{S_*}] \models \eta_j(x,y) \quad \text{ if and only if } \quad \mathcal{B}, [x{\mapsto}\mathfrak{b}^{(1)}_T, y{\mapsto}\mathfrak{b}^{(2)}_{S'}] \models \eta_j(x,y) \ .$$

Therefore, we get $\mathcal{B}_{-S_*}, [x{\mapsto}\mathfrak{b}^{(1)}_T] \models \exists y.\, \eta_j(x,y)$, which constitutes a contradiction.

This means, the number κ of unary predicate symbols occurring in ψ_{FO^2} cannot be polynomial in n, for otherwise we get $\mathcal{B}_{-S_*} \models \psi_{\mathrm{FO}^2}$ and $\mathcal{A}_{-S_*} \models \psi_{\mathrm{FO}^2}$. Since $\kappa \in \mathcal{O}\big(\mathrm{len}(\varphi_{\mathrm{FO}^2})\big)$, it follows that $\mathrm{len}(\varphi_{\mathrm{FO}^2})$ cannot be polynomial in n but must be at least exponential, in order to satisfy $2^{2^n} \leq 2^{2(n+1)+\kappa}$ for growing n. □

3.13 Separateness and Fluted Formulas

The main characteristic of fluted formulas can be crisply described as follows: "the order of quantification of variables coincides with the order in which those variables appear as arguments of predicates" ([PST16], page 1). The sentences in Herzig's ordered fragment can in fact be described in the same intuitive way. Herzig has put it like this: "the ordering of the quantifiers must be that of the variables in the predicates they govern" ([Her90], page 1). Nonetheless, the two fragments differ syntactically, as the details of their respective definition differ.

Example 3.13.1. *The following FL sentence constitutes a definition of the concept* married couples all whose children are married — $\mathrm{mwmc}(x_1,x_2)$*:*

$$\forall x_1 x_2.\, \mathrm{mwmc}(x_1,x_2) \leftrightarrow \mathrm{married}(x_1,x_2) \wedge \big(\forall x_3.\, \mathrm{haveChild}(x_1,x_2,x_3) \rightarrow \exists x_4.\, \mathrm{married}(x_3,x_4)\big) \ .$$

This exemplary sentence is taken from [HSG04]. It belongs to FL but not to Herzig's ordered fragment. The reason is that the atom $\mathrm{married}(x_3,x_4)$ *contains the variables* x_3, x_4 *whose quantifiers lie within the scope of* $\forall x_1$ *and* $\forall x_2$*, but neither* x_1 *nor* x_2 *occur (left of* x_3, x_4*) in the atom* $\mathrm{married}(x_3,x_4)$*.*

The following sentence lies in the intersection of FL and Herzig's ordered fragment. It defines the concept of married couples that do not have any children together — $\mathrm{mwoc}(x_1,x_2)$*:*

$$\forall x_1 x_2.\, \mathrm{mwoc}(x_1,x_2) \leftrightarrow \mathrm{married}(x_1,x_2) \wedge \neg\exists x_3.\, \mathrm{haveChild}(x_1,x_2,x_3) \ .$$

A simple variation of this sentence, however, does not satisfy the syntactic restrictions of FL, while it still falls into Herzig's ordered fragment:

$$\forall x_1 x_2.\, \mathrm{mwoc}(x_1,x_2) \leftrightarrow \forall x_3.\, \mathrm{married}(x_1,x_2) \wedge \neg\mathrm{haveChild}(x_1,x_2,x_3) \ .$$

As the two fragments seem to be so similar, one could ask whether they are equivalent in expressiveness. Indeed, using the concept of separateness of variables, we can reconcile the two fragments while, at the same time, extending both of them to a common superclass, called the *separated fluted fragment (SFL)*.

In the first-order formulas in this section we allow bound variables to reappear in distinct occurrences of quantifiers in the same formula. Before we formulate the definition of SFL, we adapt the following notation from the definition of Maslov's fragment K (cf. page 25). Let $\psi(u_1,\ldots,u_m)$ be any subformula of a first-order sentence φ. We assume that $u_1,\ldots,u_m$ are exactly the variables occurring freely in ψ and that they are pairwise distinct. The *φ-prefix of* ψ is the sequence $\mathcal{Q}_1 v_1 \ldots \mathcal{Q}_m v_m$ of quantifiers in φ (read from left to right) that bind the free variables of ψ, in particular, we have $\{v_1,\ldots,v_m\} = \{u_1,\ldots,u_m\}$.

$\mathcal{V}_i$

Definition 3.13.2 (Separated fluted fragment (SFL)). *Let $\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3, \ldots$ be disjoint ordered sequences of pairwise distinct variables $\mathcal{V}_i = x_1^i, x_2^i, x_3^i, \ldots$. In what follows, we occasionally treat the sequences $\mathcal{V}_i$ as sets.*

The separated fluted fragment (SFL) *comprises all relational first-order sentences φ without equality in which every atom A satisfies the following properties.*

(a) A is of the form $P(x_\ell^i, \ldots, x_k^i)$ for some predicate symbol P and certain integers i, k, ℓ with $i \geq 1$, $k \geq 0$, and $1 \leq \ell \leq k$.

(b) The φ-prefix of A is of the form $\mathcal{Q}_\ell x_\ell^i, \ldots, \mathcal{Q}_k x_k^i$ with $\mathcal{Q}_j \in \{\exists, \forall\}$.

Although separateness is not explicitly mentioned in the definition of SFL, it implicitly plays an important role. For every atom A in any SFL sentence φ, we find one sequence $\mathcal{V}_i$ from which all variables in A stem, i.e. $\mathrm{vars}(A) \subseteq \mathcal{V}_i$. Since the $\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3, \ldots$ are pairwise disjoint, they are, hence, also pairwise separated in φ.

It is not hard to see that every FL sentence also belongs to SFL. The simple monadic sentence

$$\forall x_1^1 \exists x_1^2.\, P(x_1^1) \leftrightarrow Q(x_1^2)$$

is neither fluted nor does it belong to Herzig's fluted fragment. However, it belongs to SFL. Indeed, every MFO sentence can be turned into an SFL sentence by renaming bound variables. Consider any MFO sentence φ and suppose that all quantifiers in φ bind distinct variables. Let $u_1, \ldots, u_k$ be an enumeration of all the first-order variables occurring in φ. Let φ' be the sentence that results from φ by renaming every u_i into x_1^i. This sentence φ' clearly belongs to SFL.

Finally, consider any sentence ψ that belongs to Herzig's ordered fragment. Let $P(u_1, \ldots, u_m)$ and $Q(v_1, \ldots, v_{m'})$ be two atoms in ψ. Let j, j' be any two indizes with $1 \leq j \leq m$ and $1 \leq j' \leq m'$ such that u_j and $v_{j'}$ are bound by the same quantifier $\mathcal{Q}_j u_j = \mathcal{Q}'_{j'} v_{j'}$ in ψ. By definition of Herzig's ordered fragment, these quantifiers $\mathcal{Q}_j u_j$ and $\mathcal{Q}'_{j'} v_{j'}$ are exactly in the scopes of $\mathcal{Q}_1 u_1, \ldots, \mathcal{Q}_{j-1} u_{j-1}$ and $\mathcal{Q}'_1 v_1, \ldots, \mathcal{Q}'_{j'-1} v_{j'-1}$, respectively, and no other quantifier scopes. As the quantifiers $\mathcal{Q}_j u_j$ and $\mathcal{Q}'_{j'} v_{j'}$ coincide, the sets $\{u_1, \ldots, u_j\}$ and $\{v_1, \ldots, v_{j'}\}$ must be equal. Applying this argument iteratively, we infer $j = j'$ and that the sequences $u_1, \ldots, u_j$ and $v_1, \ldots, v_{j'}$ coincide. Suppose $j_* \geq 1$ is the maximal index such that u_{j_*} and v_{j_*} are bound by the same quantifier. For any indices $\ell, \ell' > j_*$ we have that neither of the quantifiers $\mathcal{Q} u_\ell$ and $\mathcal{Q}' v_{\ell'}$ binding the variables u_ℓ and $v_{\ell'}$, respectively, lies in the scope of the other. For otherwise, assume that $\mathcal{Q} u_\ell$ were in the scope of $\mathcal{Q}' v_{\ell'}$. Hence, $\ell' < \ell$ and there is some $u_{\ell''}$ with $\ell'' < \ell$ such that $u_{\ell''}$ is also bound by the quantifier $\mathcal{Q}' v_{\ell'}$. By the above argument, we have that $\ell' = \ell''$ and that the sequences $u_1, \ldots, u_{\ell'}$ and $v_1, \ldots, v_{\ell'}$ must coincide. But since j_* is maximal and $j_* < \ell'$, we get a contradiction. Consequently, we can rename the bound variables in ψ in such a way that every atom A has the form $P(x_1^1, \ldots, x_k^1)$ for some k and the ψ-prefix of A is of the form $\mathcal{Q}_1 x_1^1, \ldots, \mathcal{Q}_k x_k^1$.

Proposition 3.13.3. *SFL properly contains (modulo renaming of bound variables) FL, MFO, and Herzig's ordered fragment.*

The following lemma stipulates that every SFL sentence has an equivalent in FL. As usual, this result is established by giving an effective equivalence-preserving translation from SFL into FL.

Lemma 3.13.4. *Every SFL sentence is equivalent to some FL sentence.*

Proof. As a primer we adapt some notation from the definition of the fluted fragment (cf. page 26). For every nonnegative integer k and every positive integer i we define the set $\mathrm{FL}^{(k)}(\mathcal{V}_i)$ inductively as follows. Any atom $P(x_\ell^i, \ldots, x_k^i)$ with $1 \leq \ell \leq k$ belongs to $\mathrm{FL}^{(k)}(\mathcal{V}_i)$. The set $\mathrm{FL}^{(k)}(\mathcal{V}_i)$ is closed under Boolean combinations, i.e. if φ and ψ belong to $\mathrm{FL}^{(k)}(\mathcal{V}_i)$, then so do $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \to \psi$, $\varphi \leftrightarrow \psi$. Given any $\mathrm{FL}^{(k+1)}(\mathcal{V}_i)$ formula $\varphi(x_\ell^i, \ldots, x_{k+1}^i)$, then $\forall x_{k+1}.\, \varphi$ and $\exists x_{k+1}.\, \varphi$ belong to $\mathrm{FL}^{(k)}(\mathcal{V}_i)$.

Consider any SFL sentence φ and let m be the smallest integer such that $\mathrm{vars}(\varphi) \subseteq \mathcal{V}_1 \cup \ldots \cup \mathcal{V}_m$. Then, all $\mathcal{V}_1, \ldots, \mathcal{V}_m$ are pairwise separated in φ. Without loss of generality, we assume that φ is in negation normal form.

We prove an auxiliary result that is an adapted version of Claim I from the proof of Lemma 3.12.3.

Claim I: Consider any subformula $\psi = \mathcal{Q}x^i_k.\,\chi$ of φ with $\mathcal{Q} \in \{\forall, \exists\}$ that satisfies the following properties:

(a) χ is a Boolean combination of formulas from $\bigcup_{k',i'} \mathrm{FL}^{(k')}(\mathcal{V}_{i'})$ — which we shall call *basic formulas* in what follows;

(b) each of these basic formulas that contains x^i_k is an $\mathrm{FL}^{(k)}(\mathcal{V}_i)$ formula;

(c) every subformula of χ that is of the form $\mathcal{Q}''x^i_k.\,\chi''$ is an $\mathrm{FL}^{(k-1)}(\mathcal{V}_i)$ formula.

basic formulas

Then, we can construct a formula ψ' such that

(1) ψ' is equivalent to ψ,

(2) ψ' is a Boolean combination of formulas from $\bigcup_{k',i'} \mathrm{FL}^{(k')}(\mathcal{V}_{i'})$, and

(3) every subformula $\mathcal{Q}x^i_k.\,\chi'$ occurring in ψ' belongs to $\mathrm{FL}^{(k-1)}(\mathcal{V}_i)$.

Proof: We treat the case where $\mathcal{Q}$ is an existential quantifier; the case of $\mathcal{Q} = \forall$ can be treated dually.

First, we transform χ into an equivalent disjunction of conjunctions of basic formulas that is of the form

$$\bigvee_j \eta_{j,i,k}(x^i_1, \ldots, x^i_k) \wedge \bigwedge_{1 \leq i' \leq m} \bigwedge_{k'} \eta'_{j,i',k'}(x^{i'}_1, \ldots, x^{i'}_{k'}) \ ,$$

where we group the basic formulas in accordance with their belonging to the sets $\mathrm{FL}^{(k')}(\mathcal{V}_{i'})$. More precisely, the conjunctions $\eta_{j,i,k}$ contain exactly those basic formulas from the j-th disjunct in which the variable bound by $\mathcal{Q}x^i_k$ occurs freely. Moreover, any basic formula from $\mathrm{FL}^{(k')}(\mathcal{V}_{i'})$ that occurs in the j-th disjunct and does not contain x^i_k as free variable is a conjunct of $\eta'_{j,i',k'}$. By assumption, each $\eta_{j,i,k}$ belongs to $\mathrm{FL}^{(k)}(\mathcal{V}_i)$, as we assumed that every basic formula in which x^i_k occurs is an $\mathrm{FL}^{(k)}(\mathcal{V}_i)$ formula.

Hence, ψ is equivalent to a formula of the form

$$\exists x^i_k.\, \bigvee_j \eta_{j,i,k}(x^i_1, \ldots, x^i_k) \wedge \bigwedge_{1 \leq i' \leq m} \bigwedge_{k'} \eta_{j,i',k'}(x^{i'}_1, \ldots, x^{i'}_{k'}) \ .$$

We shift the existential quantifier $\exists x^i_k$ inwards so that it only binds the (sub-)conjunctions $\eta_{j,i,k}$. The emerging subformula $\exists x^i_k.\,\eta_{j,i,k}$ belongs to $\mathrm{FL}^{(k-1)}(\mathcal{V}_i)$. The result

$$\bigvee_j \Big(\exists x^i_k.\, \eta_{j,i,k}(x^i_1, \ldots, x^i_k)\Big) \wedge \bigwedge_{1 \leq i' \leq m} \bigwedge_{k'} \eta_{j,i',k'}(x^{i'}_1, \ldots, x^{i'}_{k'})$$

is the sought ψ' that is a Boolean combination of formulas from $\bigcup_{k',i'} \mathrm{FL}^{(k')}(\mathcal{V}_{i'})$. ◊

By Definition 3.13.2, every atom in φ is an $\mathrm{FL}^{(k')}(\mathcal{V}_{i'})$ formula for certain k', i'. Hence, every subformula $\mathcal{Q}x^i_k.\,\chi$ of φ with quantifier-free χ satisfies the conditions of Claim I. Consider any subformula $\psi := \mathcal{Q}x^i_k.\,\chi$ of φ such that χ is a Boolean combination of atoms and of formulas $\psi' := \mathcal{Q}'x^{i'}_{k'}.\,\chi'$ that satisfy the preconditions of Claim I. By Claim I, we can transform all these ψ' into equivalent formulas ψ'' in such a way that ψ, after all these transformations, satisfies the preconditions of Claim I. Due to this observation, we can iteratively apply Claim I to transform the sentence φ into an equivalent sentence φ' that is a Boolean combination of sentences from $\bigcup_{k',i'} \mathrm{FL}^{(k')}(\mathcal{V}_{i'})$. Since every sentence $\chi \in \mathrm{FL}^{(k')}(\mathcal{V}_{i'})$ is equivalent to the sentence $\forall x^{i'}_1 \ldots x^{i'}_{k'}.\,\chi$, we can transform φ' into an equivalent sentence φ'' that is a Boolean combination of sentences from $\bigcup_{i'} \mathrm{FL}^{(0)}(\mathcal{V}_{i'})$. In φ'' the sets $\mathcal{V}_1, \ldots, \mathcal{V}_m$ are pairwise strictly separated. Hence, we can rename bound variables in φ'' in such a way that the result φ''' is a Boolean combination of sentences from $\mathrm{FL}^{(0)}(\mathcal{V}_1)$. This sentence φ''' belongs to the fluted fragment. □

Since FL enjoys the finite model property [PST16], Lemma 3.13.4 implies that the same holds true for SFL. Hence, the satisfiability problem associated with SFL (*SFL-Sat*) is decidable.

SFL-Sat

Theorem 3.13.5. *SFL possess the finite model property and, hence, the satisfiability problem for SFL sentences is decidable.*

3.14 Decidable Fragments with Function Symbols

In the previous sections our focus was relational first-order formulas, and function symbols have only played a marginal role. In the present section, we shall briefly consider first-order fragments that admit function symbols under certain constraints but still have decidable satisfiability problems.

3.14.1 Unary Functions in Arguments of Monadic Atoms

Our first observations are inspired by the step from MFO to the Löb–Gurevich fragment — MFO with unary function symbols, cf. page 23. Adopting a method already used by Löb in [Löb67] and also by Grädel (cf. proof of Proposition 6.2.7 in [BGG97]), we can handle unary function symbols under certain restrictions.

Proposition 3.14.1. *Let φ be a first-order sentence without non-unary function symbols (constant symbols are admitted). If the unary function symbols exclusively occur in atoms starting with a unary predicate symbol, then we can find an equisatisfiable sentence φ' without non-constant function symbols such that any model $\mathcal{B}$ of φ' can be transformed into a model $\mathcal{A}$ of φ over the same domain. The length of φ' lies in $\mathcal{O}\big(\mathrm{len}(\varphi)\big)$.*

Proof. The proof is an adaptation of the proof of Proposition 6.2.7 from [BGG97].

Let $f_1, \ldots, f_k$ be an enumeration of all unary function symbols that occur in φ. We apply the following transformation iteratively. Assume φ contains the atom $P(f_i(t))$ for some term t. We transform φ into $\varphi\big[P(f_i(t)) \,/\, R(t)\big] \wedge \forall x.\, P(f_i(x)) \leftrightarrow R(x)$, where the R is a fresh unary predicate symbol and $\varphi\big[P(f_i(t)) \,/\, R(t)\big]$ is the formula we obtain from φ by replacing every occurrence of $P(f_i(t))$ by $R(t)$. Starting from φ, exhaustive application of this transformation yields a sentence φ'' of the form $\psi \wedge \bigwedge_{i=1}^{k} \bigwedge_j \forall x.\, \big(P_j(f_i(x)) \leftrightarrow R_{i,j}(x)\big)$, where ψ does not contain any of the f_i anymore. If we conceive the f_i in φ' as Skolem functions and revert the Skolemization, the f_i vanish completely and we end up with the equisatisfiable sentence $\varphi' := \psi \wedge \forall x \exists y_1 \ldots y_k.\, \bigwedge_{i=1}^{k} \bigwedge_j \big(P_j(y_i) \leftrightarrow R_{i,j}(x)\big)$.

Because of $\mathrm{len}(\psi) \leq \mathrm{len}(\varphi)$ and since for any occurrence of an f_i in φ at most one new conjunct of a fixed length is introduced, we get $\mathrm{len}(\varphi') \in \mathcal{O}(\mathrm{len}(\varphi))$. □

The construction used in the proof of Lemma 3.14.1 only requires that the unary function symbols exclusively occur in the arguments of unary predicate symbols. It is not required that all occurring predicate symbols are unary. If we were to consider, for instance, an SF sentence in which unary function symbols occur in the arguments of unary predicate symbols and nowhere else, then the sentence $\psi \wedge \forall x \exists y_1 \ldots y_k.\, \bigwedge_{i=1}^{k} \bigwedge_j \big(P_j(y_i) \leftrightarrow R_{i,j}(x)\big)$ belongs to SF as well (after shifting all quantifiers to the front).

Similarly, the construction can be applied to GBSR sentences with unary function symbols in monadic atoms. Although the final result is, technically, not in GBSR, it can easily be converted to GBSR by shifting quantifiers. The same holds true for GAF, GGKS, and SFO^2. Even for the generalized guarded fragments SGF, SLGF, and SGNFO the construction is applicable, if we add trivial guards in the spirit of Propositions 3.10.4 and 3.11.3. For SFL only renaming bound variables is necessary to obtain an SFL sentence in the end. Hence, all of the mentioned fragments do not only (almost) contain MFO, but could be extended so that they (almost) become a proper superset of the Löb–Gurevich fragment.

3.14.2 SF and GBSR with Stratified Occurrences of Function Symbols

Decidable extensions of BSR with non-constant function symbols have been investigated mostly in the realm of sorted logic. Two examples are Abadi et al.'s *stratified vocabularies* [ARS07, ARS10] and Korovin's *non-cyclic sorts* [Kor13b]. A third approach is developed in [GdM09], where sorts are not the primary source of finiteness of the set of relevant terms. The cause for this limitation is rather the syntactic structure of the formula at hand. Technically, the same effect could be realized by extracting implicit sort information from the occurrences of function symbols in certain

argument positions. In the end, it is not surprising that all three approaches (almost) lead to the same fragment: *BSR with function symbols that occur only is a stratified fashion.*

Definition 3.14.2 (GBSR with stratified occurrences of function symbols)**.** *Consider any vocabulary $\Sigma = \langle \Pi, \Omega \rangle$ and let φ be a Σ-sentence that adheres to the requirements of GBSR with the exception that we allow function symbols to occur. The sentence φ is considered to be a* GBSR sentence with stratified occurrences of function symbols *if there is a mapping* $\mathrm{lvl}_\varphi : (\Pi \cup \Omega) \times \mathbb{N} \to \mathbb{N}$ *that maps* argument positions*, i.e. pairs of the form $\langle P, k \rangle$ with $P \in \Pi$ and $1 \leq k \leq \mathrm{arity}(P)$* lvl_φ *or $\langle f, k \rangle$ with $f \in \Omega$ and $1 \leq k \leq \mathrm{arity}(f) + 1$, to nonnegative integers such that the following conditions are satisfied.*

(a) For every m-ary function symbol $f \in \Omega$ and every i with $1 \leq i \leq m$ we have $\mathrm{lvl}_\varphi\langle f, i \rangle > \mathrm{lvl}_\varphi\langle f, m+1 \rangle$.

(b) For every (sub)term $g(s_1, \ldots, s_{k-1}, f(t_1, \ldots, t_m), s_{k+1}, \ldots, s_{m'})$ occurring in φ we have $\mathrm{lvl}_\varphi\langle f, m+1 \rangle = \mathrm{lvl}_\varphi\langle g, k \rangle$. This includes the case where f is a constant symbol and $m = 0$. Moreover, this also includes the case where g is replaced with a predicate symbol P.

(c) For every equation $f(s_1, \ldots, s_m) \approx g(t_1, \ldots, t_{m'})$ occurring in φ we have $\mathrm{lvl}_\varphi\langle f, m+1 \rangle = \mathrm{lvl}_\varphi\langle g, m'+1 \rangle$. This includes the cases where f or g or both are constant symbols (with $m = 0$ or $m' = 0$ or both, respectively).

(d) Every variable v that occurs in φ is associated with a fixed nonnegative integer ℓ_v such that we have the following ℓ_v

- *for every (sub)term $f(s_1, \ldots, s_{k-1}, v, s_{k+1}, \ldots, s_m)$ in φ we have $\mathrm{lvl}_\varphi\langle f, k \rangle = \ell_v$,*
- *for every atom $P(s_1, \ldots, s_{k-1}, v, s_{k+1}, \ldots, s_m)$ in φ we have $\mathrm{lvl}_\varphi\langle P, k \rangle = \ell_v$,*
- *for every equation $v \approx g(t_1, \ldots, t_m)$ in φ we have $\mathrm{lvl}_\varphi\langle g, m+1 \rangle = \ell_v$, and*
- *for every equation $v \approx v'$ we have $\ell_v = \ell_{v'}$.*

Intuitively, the main characteristic of collections of terms in which function symbols only occur in a stratified fashion is that any function symbol f does never occur directly or indirectly in the arguments applied to f. An example for a direct occurrence is $f(\bar{s}_1, g(\bar{t}_1, f(\bar{t}'), \bar{t}_2), \bar{s}_2)$, in which f occurs in an argument in a term $f(\ldots)$. Indirect occurrences require, for instance, two terms $f(\bar{s}_1, g(\bar{t}), \bar{s}_2)$ and $g(\bar{s}'_1, f(\bar{t}'), \bar{s}'_2)$ where f does not occur in $\bar{s}_1, \bar{s}_2, \bar{t}, \bar{s}'_1, \bar{s}'_2, \bar{t}'$.

Definition 3.14.2 resembles an a-posteriori variant of the definition of *stratified vocabulary* defined in [ARS10] for a sorted setting.

Definition 3.14.3 (Stratified vocabulary — Definition 1 from [ARS10])**.** *A vocabulary Σ for many-sorted logic is* stratified *if there is a mapping lvl_Σ from sorts to nonnegative integers such that for every function symbol $f : \xi_1 \times \ldots \times \xi_m \to \xi_{m+1}$ we have $\mathrm{lvl}_\Sigma(\xi_i) > \mathrm{lvl}_\Sigma(\xi_{m+1})$ for every $i \leq m$.*

Stratified vocabularies are essentially the same objects as *non-cyclic vocabularies*, which are defined and investigated in [Kor13b]. A conceptually different approach, which yet leads to essentially the same first-order fragment is developed in [GdM09], Section 3. Instead of the mapping lvl, the authors use set constraints for the analysis of the syntactic structure.

Proposition 3.14.4 ([ARS10, GdM09, Kor13b])**.** *The satisfiability problem for multi-sorted $\exists^*\forall^*$-sentences over a stratified vocabulary is decidable.*

The main argument for proving Proposition 3.14.4 is that any Herbrand domain over many-sorted stratified vocabularies is finite. Exhaustive Skolemization of an $\exists^*\forall^*$-sentence φ over a stratified vocabulary only introduces constant symbols. As this again leads to a stratified vocabulary, Lemma 1.0.4 entails that any Skolemized version of φ has a finite Herbrand model, if φ is satisfiable. In other words, the class of many-sorted $\exists^*\forall^*$-sentences over a stratified vocabulary enjoys the finite model property.

Although Definition 3.14.2 does not rely on sort information, it ensures that a formula is constructed in accordance with Definition 3.14.3, based on implicit sort information that can be reconstructed a posteriori by an analysis of the occurrences of function symbols in terms.

Proposition 3.14.5. *Consider any Σ-formula φ that satisfies Definition 3.14.2. Then, the single-sorted vocabulary Σ can be turned into a many-sorted vocabulary Σ' such that (a) Σ' and Σ contain the same function and predicate symbols, (b) Σ' is stratified, and (c) φ is a Σ'-formula obeying the sort restrictions of Σ'.*

We have already pointed out earlier that the translations from SF and GBSR into BSR that underly Lemmas 3.2.5 and 3.5.2 also works in the presence of function symbols. This also entails that every GBSR sentence with function symbols satisfying Definition 3.14.2 is equivalent to some BSR sentence that also satisfies this definition. Hence, Proposition 3.14.4 entails that the fragment described in Definition 3.14.2 has a decidable satisfiability problem

Theorem 3.14.6. *The satisfiability problem for GBSR sentences with stratified occurrences of function symbols is decidable.*

3.14.3 Monadic Horn Sentences in which Positive Literals are Shallow and Linear

The *monadic shallow linear Horn fragment (MSLH)* (see page 27 for the exact definition) is quite different from the other fragments we have treated so far. It has strong connections to certain kinds of tree automata, see, e.g. [JMW98] and [Wei98], Section 4. MSLH can be conceived as an extension of the class of Horn MFO sentences after exhaustive Skolemization. Given any Horn MFO sentence $\exists z \forall x_1 \exists y_1 \ldots \forall x_n \exists y_n .\ \bigwedge_i C_i(\bar{\mathrm{x}}, \bar{\mathrm{y}})$, exhaustive Skolemization yields

$$\forall x_1 \ldots x_n .\ \bigwedge_i C_i\big[z/c, y_1/f_1(x_1), \ldots, y_n/f_n(x_1, \ldots, x_n)\big] \ ,$$

which belongs to MSLH, as every positive literal has one of three shapes: $P(c)$, $P(x_i)$, or $P(f_i(x_1, \ldots, x_i))$ with pairwise distinct $x_1, \ldots, x_i$. On the other hand, it is easy to see that MSLH allows a much richer term syntax than Skolemized MFO. For instance, terms in negative literals may be arbitrarily complicated, including nested occurrences of non-constant function symbols. Moreover, the order of variables may vary in distinct literals, whereas in Skolemized MFO the arguments in Skolem terms adhere to a fixed order. Certain forms of symmetry can be expressed in MSLH, which cannot be expressed in MFO, for example $\forall xy .\ P(f(x, y)) \rightarrow P(f(y, x))$.

The satisfiability problem for MSLH is known to be decidable in deterministic exponential time, in fact, it is ExpTime-complete [Gou05]. This also entails that the satisfiability problem for Horn MFO sentences lies in ExpTime. Although it is already known that the satisfiability problem for Horn MFO sentences is also ExpTime-hard [DL84b], it is instructive to show this by a reduction to a basic problem over tree automata. This will highlight the close connection between the two formalisms.

Proposition 3.14.7. *The satisfiability problem for Horn-MFO is* ExpTime*-complete.*

Proof sketch. To derive the upper bound, we recall that exhaustively Skolemized MFO sentences belong to the MSLH fragment for which a decision procedure is known that runs in deterministic exponential time ([Gou05], Theorem 6).

The lower bound can be derived by a reduction to the *intersection non-emptiness problem* for deterministic tree automata: Given a finite sequence of tree automata $\mathfrak{A}_1, \ldots, \mathfrak{A}_m$, is there at least one tree that is accepted by all $\mathfrak{A}_i$. This problem is known to be complete for ExpTime (see, e.g., [CDG$^+$08], Theorem 1.7.5).[6]

[6] ExpTime-hardness of the intersection non-emptiness problem was already pointed out in [FSVY91] accompanied by a very brief proof sketch. A detailed proof for the case of (bottom-up) deterministic tree automata is given in [Vea97a] and [Vea97b], Lemma 5.4. More references and historical background can be found in [Vea97a] and in the bibliographic notes in Section 1.9 of [CDG$^+$08].

We stick to the basic definition from [Vea97a]. A *tree automaton* $\mathfrak{A}$ is a quadruple $\mathfrak{A} = \langle Q, \Omega, R, F\rangle$, where *tree automaton* $\mathfrak{A}$

Q is a finite set of states;

Ω a set of function symbols, each equipped with a fixed arity $m_f \geq 0$; we require $m_f = 0$ for at least one $f \in \Omega$;

R is the transition relation, containing transition rules of the form $f(q_1, \ldots, q_{m_f}) \to q$ with $f \in \Omega$ and $q_1, \ldots, q_{m_f}, q \in Q$; and

$F \subseteq Q$ is the set of final states.

We assume that the sets Q and Ω are disjoint. A tree automaton $\mathfrak{A} = \langle Q, \Omega, R, F\rangle$ is *deterministic* if no rules in R have identical left-hand sides. Given $\mathfrak{A}$, the underlying *tree language* is the set of all syntax trees of ground terms over the function and constant symbols in Ω. The rules in R can be conceived a rewriting rules that turn (sub)trees into states. The starting point of this process are the leaves of a tree, which are represented by constant symbols from Ω. Given $c \in \Omega$, a rewrite rule starting from c looks like $c() \to q$ — we usually drop the empty list of arguments () for convenience. A term t is *accepted* by $\mathfrak{A}$ if repeated application of the rewrite rules from R eventually turns t into a final state $q \in F$. For example, consider the term $t = g(c, f(c, d))$ and suppose the rules $c \to q_1$; $d \to q_2$; $f(q_1, q_2) \to q_1$; $g(q_1, q_1) \to q_2$ belong to R. Then t is rewritten as follows *tree language accepted by* $\mathfrak{A}$

$$g(c, f(c, d)) \xrightarrow{\mathfrak{A}}{}^{+} g(q_1, f(q_1, q_2)) \xrightarrow{\mathfrak{A}} g(q_1, q_1) \xrightarrow{\mathfrak{A}} q_2 \ .$$

The term t is accepted by $\mathfrak{A}$ if and only if $q_2 \in F$.

Let $\mathfrak{A}_1, \ldots, \mathfrak{A}_n$ be a sequence of tree automata $\mathfrak{A}_i := \langle Q_i, \Omega, R_i, F_i\rangle$ that share the underlying tree vocabulary. Without loss of generality we assume that the sets $Q_1, \ldots, Q_n$ are pairwise disjoint. Let φ_1 be the following sentence

$$\varphi_1 := \bigwedge_{f\in\Omega} \forall x_1 \ldots x_{m_f}. \bigwedge_{i=1}^{n} \bigwedge_{f(q_1,\ldots q_{m_f})\to q \,\in\, R_i} \big(P_{q_1}(x_1) \wedge \ldots \wedge P_{q_{m_f}}(x_{m_f}) \to P_q(f(x_1, \ldots, x_{m_f}))\big) \ ,$$

which encodes all the rules from the transition relations of all tree automata $\mathfrak{A}_1, \ldots, \mathfrak{A}_n$. Clearly, φ_1 can easily be converted into a Horn sentence. Since each of the clauses in that Horn sentence contains some positive literal, the sentence is satisfiable. Hence, by Proposition 1.0.5, φ_1 has a unique minimal Herbrand model $\mathcal{H}_1$. It is easy to verify that the language accepted by any $\mathfrak{A}_i$ is resembled by the set $\bigcup_{q\in F_i} P_q^{\mathcal{H}_1}$. Put differently, for every ground term t that is accepted by $\mathfrak{A}_i$ we find some final state $q \in F_i$ such that $\mathcal{H}_1 \models P_q(t)$. In addition, we notice that φ_1 is a Skolemized variant of an MFO sentence that is (almost) Horn.

Next, we define the sentence φ_2 that introduces the predicate symbols $S_1, \ldots, S_n$:

$$\varphi_2 := \bigwedge_{i=1}^{n} \bigwedge_{q\in F_i} \forall x. \big(P_q(x) \to S_i(x)\big) \ .$$

For each of the S_i we observe that the unique minimal Herbrand model $\mathcal{H}_2 \models \varphi_1 \wedge \varphi_2$ interprets the S_i so that the set $S_i^{\mathcal{H}_2}$ captures the language accepted by $\mathfrak{A}_i$.

In order to also capture the intersection of the accepted languages in a single predicate, we define the sentence

$$\varphi_3 := \ \forall x. \big(S_1(x) \wedge \ldots \wedge S_n(x) \to T(x)\big) \ \wedge \ \bigwedge_{i=1}^{n} \big(T(x) \to S_i(x)\big) \ .$$

Still, the conjunction $\varphi_1 \wedge \varphi_2 \wedge \varphi_3$ is satisfiable and, hence, has a unique minimal Herbrand model $\mathcal{H}_3$. Then, we observe that the set $T^{\mathcal{H}_3}$ comprises exactly the terms that are in the intersection of the languages accepted by $\mathfrak{A}_1, \ldots, \mathfrak{A}_n$.

Finally, consider the sentence $\varphi := \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \forall x.\, \neg T(x)$. It is equivalent to the Skolemized variant of a Horn-MFO sentence and, if φ is satisfiable, it has a unique minimal Herbrand model $\mathcal{H}_*$. Such a model exists if and only if the intersection of the tree languages accepted by $\mathfrak{A}_1, \ldots, \mathfrak{A}_n$ is empty. Consequently, φ is unsatisfiable if and only if the intersection of the languages accepted by $\mathfrak{A}_1, \ldots, \mathfrak{A}_n$ is not empty. In addition, the length of φ is linear in the sum of the lengths of any reasonable representation of the tree automata $\mathfrak{A}_1, \ldots, \mathfrak{A}_n$. □

It seems that treatments of the semantics of MSLH sentences, and their close relatives from the fragment H_1 (cf. page 27), mainly consider Herbrand structures and the correspondence to tree automata. This is the case in [Wei99, NNS02, Gou05, TW15, TW17, Teu17]. Obviously, Herbrand structures for MSLH sentences with non-constant function symbols have an infinite domain. Nevertheless, these structures might be representable by finite means. In the present section we shall see how to construct models with finite domains for satisfiable MSLH sentences. Hence, we show that this fragment enjoys the finite model property. More precisely, every satisfiable MSLH sentence containing k constant symbols, p predicate symbols, and function symbols of arity at most m has a model $\mathcal{B}$ whose domain contains at most $k + (m+1) \cdot 2^p$ elements. The finite model $\mathcal{B}$ we shall construct can be conceived as a finite representation of the *minimal Herbrand model* $\mathcal{H}$ — every element in $\mathcal{B}$'s domain corresponds to an equivalence class over ground terms in $\mathcal{H}$'s domain. The underlying equivalence relation is determined by the interpretation of the unary predicate symbols under $\mathcal{H}$.

N, φ Fix some vocabulary $\Sigma = \langle \Pi, \Omega \rangle$ and consider a finite set N of pairwise variable-disjoint Σ-clauses for which the sentence $\varphi := \forall \bar{\mathrm{x}}.\, \bigwedge_{C \in N} C(\bar{\mathrm{x}})$ is satisfiable and belongs to the MSLH fragment. The following is an immediate consequence of Lemma 4 from [Wei99].

Proposition 3.14.8. *There is a finite set N_* of pairwise variable-disjoint Σ-clauses such that $N \subseteq N_*$ and the sentence $\varphi_* := \forall \bar{\mathrm{x}}'.\, \bigwedge_{C \in N_*} C(\bar{\mathrm{x}}')$ belongs to MSLH and is logically entailed by φ. Moreover, there is some Herbrand model $\mathcal{H} \models \varphi_*$ such that for every ground Σ-atom A of the form $S\big(f(s_1, \ldots, s_m)\big)$ we have $\mathcal{H} \models A$ only if there is some clause C in N_* and a variable assignment β that satisfy the following properties:*

(a) C has the form $\neg P_1(x_1) \vee \ldots \vee \neg P_n(x_n) \vee S\big(f(y_1, \ldots, y_m)\big)$ where $\{x_1, \ldots, x_n\} \subseteq \{y_1, \ldots, y_m\}$ and $f(y_1, \ldots, y_m)$ is linear, i.e. the $y_1, \ldots, y_m$ are pairwise distinct; $n = 0$ or $m = 0$ is allowed;

(b) we have $\beta(y_i) = s_i$ for every i, $1 \leq i \leq m$; and

(c) we have $\mathcal{H}, \beta \models P_j(x_j)$ for every j, $1 \leq j \leq n$.

Since φ^* is Horn and satisfiable, Proposition 1.0.5 entails that it possesses a unique minimal Herbrand model $\mathcal{H}$. The property described in Proposition 3.14.8 provides the key to construct a finite model for φ. The following example is intended to illustrate the underlying ideas.

Example 3.14.9. *Consider the following set of clauses:*

$$\begin{aligned} N := \{ & P(a),\ Q(b), \\ & \neg P(u) \vee \neg P(u') \vee P(f(u, u')), \\ & \neg Q(v) \vee \neg Q(v') \vee Q(f(v, v')), \\ & \neg P(x) \vee R(f(x, y)), \\ & \neg P(y) \vee R(f(x, y)), \\ & \neg Q(x) \vee R(f(x, y)), \\ & \neg Q(y) \vee R(f(x, y)), \\ & \neg P(z) \vee \neg Q(z) \vee \neg R(z)\} \end{aligned}$$

where a and b are constant symbols. The sentence $\varphi := \forall u u' v v' x y z.\, \bigwedge_{C \in N} C(x, y, z)$ is satisfied by

the Herbrand structure $\mathcal{H}$ with

$$P^{\mathcal{H}} = \{a, f(a,a), f(a,f(a,a)), f(f(a,a),a), f(f(a,a),f(a,a)), f(a,f(a,f(a,a))), \ldots\},$$
$$Q^{\mathcal{H}} = \{b, f(b,b), f(b,f(b,b)), f(f(b,b),b), f(f(b,b),f(b,b)), f(b,f(b,f(b,b))), \ldots\},$$
$$R^{\mathcal{H}} = \{f(s,t) \mid s, t \text{ are any ground } \Sigma\text{-terms}\}.$$

The model $\mathcal{H}$ is not minimal in the sense that the set $R^{\mathcal{H}}$ is larger than necessary. When we fix the interpretations of P and Q under $\mathcal{H}$, the clauses in N enforce only the terms $f(s,t)$ with $s \in P^{\mathcal{H}}$ and $t \in Q^{\mathcal{H}}$ to occur in R's interpretation. In other words, the Herbrand structure $\mathcal{H}'$ with $P^{\mathcal{H}'} := P^{\mathcal{H}}$, $Q^{\mathcal{H}'} := Q^{\mathcal{H}}$, and

$$R^{\mathcal{H}'} := \{f(s,t) \mid s \in P^{\mathcal{H}} \text{ or } t \in Q^{\mathcal{H}}\}$$

is a model of φ whose interpretation of R is a proper subset of $R^{\mathcal{H}}$. In contrast to $\mathcal{H}$, the structure $\mathcal{H}'$, together with $N_* := N$, satisfies the conditions of Proposition 3.14.8: for every term $f(s,t)$ that belongs to $R^{\mathcal{H}'}$ we have that one of the clauses $\neg P(x) \vee R(f(x,y))$ or $\neg P(y) \vee R(f(x,y))$ or $\neg Q(x) \vee R(f(x,y))$ or $\neg Q(y) \vee R(f(x,y))$ enforces $\mathcal{H} \models R(f(s,t))$ because of $\mathcal{H} \models P(s)$ or $\mathcal{H} \models P(t)$ or $\mathcal{H} \models Q(s)$ or $\mathcal{H} \models Q(t)$, respectively. Similarly, the presence of any term $f(\ldots)$ in $P^{\mathcal{H}'}$ or $Q^{\mathcal{H}'}$ is enforced by one of the clauses $\neg P(u) \vee \neg P(u') \vee P(f(u,u'))$ and $\neg Q(v) \vee \neg Q(v') \vee Q(f(v,v'))$.

These requirements towards the minimality of $\mathcal{H}'$ provide us with a certain knowledge about distinct terms $f(s,t)$ and $f(s',t')$. Suppose the terms s and s' are indistinguishable with respect to their belonging to the predicates $P^{\mathcal{H}'}, Q^{\mathcal{H}'}, R^{\mathcal{H}'}$. Further suppose that the same holds for the terms t and t'. Then, $f(s,t)$ and $f(s',t')$ are also indistinguishable with respect to their belonging to $P^{\mathcal{H}'}, Q^{\mathcal{H}'}$, and $R^{\mathcal{H}'}$. A formal statement of this property is given in Lemma 3.14.10.

Based on this observation, we can use $\mathcal{H}'$ as a blueprint for a finite model $\mathcal{A}$, which is depicted in Figure 3.6. The domain of $\mathcal{A}$ shall be $\mathsf{A} := \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}\}$, and we set $a^{\mathcal{A}} := \mathsf{a}$ and $b^{\mathcal{A}} := \mathsf{b}$. The predicate symbols are interpreted by $P^{\mathcal{A}} := \{\mathsf{a}, \mathsf{c}\}, Q^{\mathcal{A}} := \{\mathsf{b}, \mathsf{d}\}, R^{\mathcal{A}} := \{\mathsf{c}, \mathsf{d}, \mathsf{e}\}$. Moreover, we define

$$f^{\mathcal{A}}(\mathsf{a},\mathsf{a}) := \mathsf{c} \qquad f^{\mathcal{A}}(\mathsf{a},\mathsf{c}) := \mathsf{c} \qquad f^{\mathcal{A}}(\mathsf{c},\mathsf{a}) := \mathsf{c} \qquad f^{\mathcal{A}}(\mathsf{c},\mathsf{c}) := \mathsf{c}$$
$$f^{\mathcal{A}}(\mathsf{b},\mathsf{b}) := \mathsf{d} \qquad f^{\mathcal{A}}(\mathsf{b},\mathsf{d}) := \mathsf{d} \qquad f^{\mathcal{A}}(\mathsf{d},\mathsf{b}) := \mathsf{d} \qquad f^{\mathcal{A}}(\mathsf{d},\mathsf{d}) := \mathsf{d}.$$

For all other inputs, $f^{\mathcal{A}}$ shall yield e as output. Every domain element in A represents one equivalence class of the terms in $\mathcal{H}'$'s Herbrand domain with respect to membership in the sets $P^{\mathcal{H}'}$, $Q^{\mathcal{H}'}$, and $R^{\mathcal{H}'}$. The domain element a represents the class $[a] := \{a\}$ of terms that belong to $P^{\mathcal{H}'}$ and to no other set. Similarly, b represents $[b] := \{b\}$ of terms that belong to $Q^{\mathcal{H}'}$ and to no other set. The element c represents the class of all terms belonging to $P^{\mathcal{H}'} \cap R^{\mathcal{H}'}$, i.e. to the class containing $f(a,a), f(a,f(a,a))$ and so on. The class of terms belonging to $Q^{\mathcal{H}'} \cap R^{\mathcal{H}'}$ is represented by d. Finally, e corresponds to the class of all terms that are member of $R^{\mathcal{H}'}$ but of none of the other predicates, e.g. $f(a,b), f(a,f(b,a))$.

We next describe formally how to construct a finite model for the given MSLH clause set N. Let N_*, φ_*, and $\mathcal{H}$ be the objects described in Proposition 3.14.8. Then, we have $\mathcal{H} \models \varphi_*$ and $\mathcal{H} \models \varphi$. Let H be the domain of $\mathcal{H}$, i.e. H is the set of all ground terms over Σ. We aim at constructing a finite model $\mathcal{B} \models \varphi = \forall \bar{\mathrm{x}}.\ \bigwedge_{C \in N} C(\bar{\mathrm{x}})$ starting from $\mathcal{H}$. N_*, φ_*, $\mathcal{H}$

Recall that Π is the set of all predicate symbols occurring in N, and that Π contains only unary predicate symbols. Let $\mathcal{P}(\Pi)$ denote the power set of Π. We define the coloring $\nu : \mathsf{H} \to \mathcal{P}(\Pi)$ such that $\nu(s) := \{P \in \Pi \mid s \in P^{\mathcal{H}}\}$ for every ground term $s \in \mathsf{H}$. Based on ν, we define the equivalence relation $\sim_\nu$ on H such that we have $s \sim_\nu t$ if and only if $\nu(s) = \nu(t)$. For every color $\mathcal{C} \subseteq \Pi$ for which H contains at least one element s with $\nu(s) = \mathcal{C}$, we pick one representative $\alpha_{\mathcal{C}} \in \mathsf{H}$ with $\nu(\alpha_{\mathcal{C}}) = \mathcal{C}$.[7] Hence, for every non-empty equivalence class $[s]_{\sim_\nu}$ in the quotient set $\mathsf{H}/_{\sim_\nu}$ we have that $[\alpha_{\nu(s)}]_{\sim_\nu} = [s]_{\sim_\nu}$ and $\nu(\alpha_{\nu(s)}) = \nu(s)$. ν, $\sim_\nu$ $\alpha_{\mathcal{C}}$

[7] Technically, this definition would generate a further domain element $\mathsf{f} \in \mathsf{A}$ for the color $\mathcal{C} = \emptyset$ in Example 3.14.9, which we have not added for simplicity.

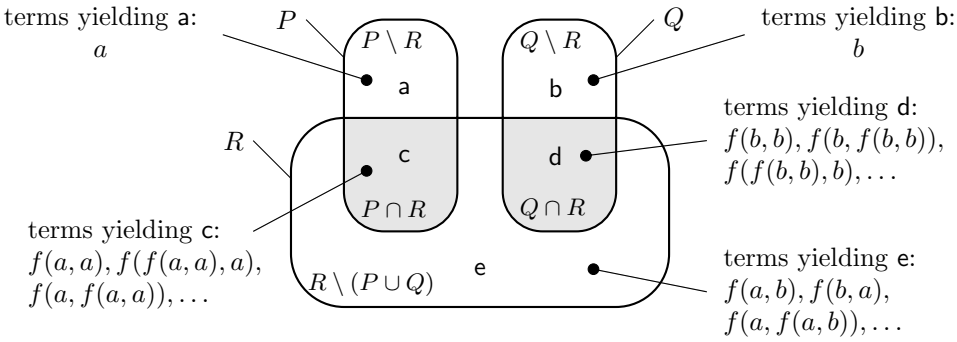


Figure 3.6: Illustration of the model $\mathcal{A}$ of φ from Example 3.14.9.

Lemma 3.14.10. *For every non-constant function symbol $f \in \Omega$ of arity m and all tuples $\langle s_1, \ldots, s_m\rangle, \langle t_1, \ldots, t_m\rangle \in \mathsf{H}^m$ for which $\nu(s_i) = \nu(t_i)$ holds for every i we have $\nu\big(f(s_1, \ldots, s_m)\big) = \nu\big(f(t_1, \ldots, t_m)\big)$.*

Proof. By Definition of $\mathcal{H}$ and ν, for every $S \in \nu\big(f(s_1, \ldots, s_m)\big)$ there is a clause C of the form $\neg P_1(x_1) \vee \ldots \vee \neg P_n(x_n) \vee S\big(f(y_1, \ldots, y_m)\big)$ in N_* and a variable assignment β that satisfy Properties (a) to (c) from Proposition 3.14.8. Let γ be a variable assignment for which we have $\gamma(y_i) := t_i$ for every i. Notice that such a γ with $\big\langle\gamma(y_1), \ldots, \gamma(y_m)\big\rangle = \langle t_1, \ldots, t_m\rangle$ always exists because the $y_1, \ldots, y_m$ are pairwise distinct. Since we assume $\nu(s_i) = \nu(t_i)$ for every i and because of $\{x_1, \ldots, x_n\} \subseteq \{y_1, \ldots, y_m\}$, Conditions (b) and (c) of Proposition 3.14.8 require $\beta(x_j) \in P_j^{\mathcal{H}}$ and, hence, we also have $\gamma(x_j) \in P_j^{\mathcal{H}}$ for every j. Since $\mathcal{H}$ is a model of φ_*, we have $\mathcal{H}, \gamma \models C$. This together with $\mathcal{H}, \gamma \models P_j(x_j)$, for every j, entails $\mathcal{H}, \gamma \models S(f(y_1, \ldots, y_m))$. Put differently, we have $S \in \nu\big(f(t_1, \ldots, t_m)\big)$.

Consequently, we obtain $\nu\big(f(s_1, \ldots, s_m)\big) \subseteq \nu\big(f(t_1, \ldots, t_m)\big)$. The converse direction can be shown by a symmetric argument. □

m_*, B_k

λ

We now construct the finite structure $\mathcal{B}$. Let m_* be the smallest positive integer such that every function symbol occurring in N has an arity of at most m_*. We define $\mathcal{B}$'s domain B to be the disjoint union of $m_* + 2$ subdomains $\mathsf{B}_0, \mathsf{B}_1, \ldots, \mathsf{B}_{m_*+1}$. Together with these subdomains we define a coloring $\lambda : \mathsf{B} \to \mathcal{P}(\Pi)$ as follows. B_0 contains exactly one element a_c for every constant symbol c occurring in N, and we require $\lambda(\mathsf{a}_c) := \nu(c)$. Moreover, we set $c^{\mathcal{B}} := \mathsf{a}_c$. This guarantees that $c^{\mathcal{B}} \neq d^{\mathcal{B}}$ for all distinct constant symbols c, d occurring in N. Let $\mathrm{im}(\nu)$ be the image of ν, i.e. $\mathrm{im}(\nu) := \{\mathcal{C} \subseteq \Pi \mid \text{there is some } s \in \mathsf{H} \text{ for which } \nu(s) = \mathcal{C}\}$. For every color $\mathcal{C} \in \mathrm{im}(\nu)$ each B_i, $1 \leq i \leq m_* + 1$, shall contain exactly one element a for which we set $\lambda(\mathsf{a}) := \mathcal{C}$.

We define the interpretation of each predicate symbol $P \in \Pi$ under $\mathcal{B}$ such that for every $\mathsf{a} \in \mathsf{B}$ we have $\mathsf{a} \in P^{\mathcal{B}}$ if and only if $P \in \lambda(\mathsf{a})$. Regarding the non-constant function symbols f occurring in N, we proceed as follows. Let m be the arity of f. Consider any tuple $\langle \mathsf{a}_1, \ldots, \mathsf{a}_m\rangle \in \mathsf{B}^m$. Let $\mathcal{C} := \nu(f(\alpha_{\lambda(\mathsf{a}_1)}, \ldots, \alpha_{\lambda(\mathsf{a}_m)}))$. Pick some index j, $1 \leq j \leq m_* + 1$, such that none of the $\mathsf{a}_1, \ldots, \mathsf{a}_m$ belongs to B_j; then $f^{\mathcal{B}}(\mathsf{a}_1, \ldots, \mathsf{a}_m) \in \mathsf{B}_j$ guarantees that $f^{\mathcal{B}}(\mathsf{a}_1, \ldots, \mathsf{a}_m) \neq \mathsf{a}_\ell$ for every ℓ, $1 \leq \ell \leq m$. Let b be the (unique) element in B_j such that $\lambda(\mathsf{b}) = \mathcal{C}$. We set $f^{\mathcal{B}}(\mathsf{a}_1, \ldots, \mathsf{a}_m) := \mathsf{b}$. This ensures $\lambda\big(f^{\mathcal{B}}(\mathsf{a}_1, \ldots, \mathsf{a}_m)\big) = \nu(f(\alpha_{\lambda(\mathsf{a}_1)}, \ldots, \alpha_{\lambda(\mathsf{a}_m)}))$.

Notice that the number of elements in $\mathcal{B}$'s domain could potentially be reduced by taking only the elements that are generated via the functions $f^{\mathcal{B}}$ starting from the elements in B_0. Then, $\mathcal{B}$ would more closely correspond to the Herbrand domain H.

Lemma 3.14.11. *Let γ be any variable assignment over $\mathcal{B}$'s domain. Let β be the variable assignment over $\mathcal{H}$'s domain defined such that for every x we have $\beta(x) := \alpha_{\lambda(\gamma(x))}$. Then, for every term t in N we have $\nu\big(\mathcal{H}(\beta)(t)\big) = \lambda\big(\mathcal{B}(\gamma)(t)\big)$.*

Proof. We proceed by induction on the structure of the term t. For the base case, assume that t is either a variable x or a constant symbol c. In the former case, we get $\nu\big(\mathcal{H}(\beta)(x)\big) = \nu\big(\beta(x)\big) = \nu\big(\alpha_{\lambda(\gamma(x))}\big) = \lambda\big(\gamma(x)\big) = \lambda\big(\mathcal{B}(\gamma)(x)\big)$. In the latter case, we get $\nu\big(\mathcal{H}(\beta)(c)\big) = \nu\big(c\big) = \lambda(\mathsf{a}_c) = \lambda\big(c^{\mathcal{B}}\big) = \lambda\big(\mathcal{B}(\gamma)(c)\big)$.

For the inductive case, assume that t is of the form $f(t_1, \ldots, t_m)$. Let $s_i := \mathcal{H}(\beta)(t_i)$ for every i. Moreover, let $\mathsf{a}_i := \mathcal{B}(\gamma)(t_i)$ for every i. By induction, we have $\nu(s_i) = \lambda(\mathsf{a}_i)$ for every i. By virtue of Lemma 3.14.10, we thus get

$$\nu\big(f(s_1, \ldots, s_m)\big) \overset{\text{L3.14.10}}{=} \nu\big(f(\alpha_{\nu(s_1)}, \ldots, \alpha_{\nu(s_m)})\big) \overset{\text{IH}}{=} \nu\big(f(\alpha_{\lambda(\mathsf{a}_1)}, \ldots, \alpha_{\lambda(\mathsf{a}_m)})\big).$$

By definition of $f^{\mathcal{B}}$, we obtain $\nu\big(f(\alpha_{\lambda(\mathsf{a}_1)}, \ldots, \alpha_{\lambda(\mathsf{a}_m)})\big) = \lambda\big(f^{\mathcal{B}}(\mathsf{a}_1, \ldots, \mathsf{a}_m)\big)$. Put together, this yields $\nu\big(f(s_1, \ldots, s_m)\big) = \lambda\big(f^{\mathcal{B}}(\mathsf{a}_1, \ldots, \mathsf{a}_m)\big)$. □

For the special case of ground terms, we can reformulate Lemma 3.14.11 into a simpler form: for every ground term t and every predicate symbol $P \in \Pi$ we have $\mathcal{H} \models P(t)$ if and only if $\mathcal{B} \models P(t)$.

Using Lemma 3.14.11, it is easy to show that φ is satisfied by the finite structure $\mathcal{B}$.

Lemma 3.14.12. *$\mathcal{B}$ is a model of φ.*

Proof. Let C be any clause in N. Since φ is satisfiable, C cannot be the empty clause. Suppose there is some variable assignment γ over $\mathcal{B}$'s domain such that $\mathcal{B}, \gamma \not\models C$. Let β be the variable assignment over $\mathcal{H}$'s domain defined by $\beta(x) := \alpha_{\lambda(\gamma(x))}$ for every x. Consider any atom $P(t)$ in C. The structure $\mathcal{B}$ is defined such that $\mathcal{B}, \gamma \models P(t)$ holds if and only if $P \in \lambda\big(\mathcal{B}(\gamma)(t)\big)$. Moreover, by definition of ν, we have $\mathcal{H}, \beta \models P(t)$ if and only if $P \in \nu\big(\mathcal{H}(\beta)(t)\big)$. Hence, Lemma 3.14.11 entails that $\mathcal{B}, \gamma \models P(t)$ holds if and only if $\mathcal{H}, \beta \models P(t)$ does. But then, $\mathcal{B}, \gamma \not\models C$ entails $\mathcal{H}, \beta \not\models C$. This contradicts our assumption that $\mathcal{H}$ is a model of φ. Hence, we must have $\mathcal{B} \models C$. □

Theorem 3.14.13. *Every satisfiable MSLH sentence φ has a finite model whose domain contains at most $k + (m+1) \cdot 2^p$ elements, where k is the number of constant symbols in φ, p the number of predicate symbols in φ, and all function symbols in φ have an arity of at most m.*

Put differently, the MSLH fragment enjoys the finite model property.

Chapter 4

The Semantic Side: Weak Dependences and Model Checking Games

In the present chapter we aim to develop a better understanding of the semantic properties of GBSR and GAF sentences. The semantic counterpart of separateness is *weak dependence*. We have already pointed out in Chapter 2, pages 19 to 20, that existentially quantified variables can in general depend on universally quantified variables. Applying standard Skolemization to the existentially quantified y in the first-order sentence $\varphi := \forall xz\exists y.\, P(x) \leftrightarrow \big(Q(x) \leftrightarrow R(y,z)\big)$, for example, leads to the replacement of every occurrence of y with the Skolem term $f(x,z)$. The result is the sentence $\varphi_{\mathrm{Sk}} := \forall xz.\, P(x) \leftrightarrow \big(Q(x) \leftrightarrow R\big(f(x,z),z\big)\big)$ in which the Skolem function f is implicitly existentially quantified. In some sense, a model $\mathcal{B}$ of φ_{Sk} must be a bit more specific than any model $\mathcal{A}$ of φ in that $\mathcal{B}$ needs to explicitly provide a mapping $f^{\mathcal{B}} : \mathsf{B} \times \mathsf{B} \to \mathsf{B}$ that returns some suitable domain element $f^{\mathcal{B}}(\mathsf{a},\mathsf{b})$, given any two elements a and b that have been assigned to x and z, respectively. We do not expect such an explicit semantic object from $\mathcal{A}$. The semantics definition merely says that every variable assignment $\beta := [x{\mapsto}\mathsf{a}, z{\mapsto}\mathsf{b}]$ can be extended to some variable assignment $\gamma := \beta[y{\mapsto}\mathsf{c}]$ such that $\mathcal{A},\gamma$ satisfy the matrix of φ, in symbols $\mathcal{A},\gamma \models P(x) \leftrightarrow \big(Q(x) \leftrightarrow R(y,z)\big)$. This means that we only *implicitly* ask for the existence of some *strategy* σ for finding suitable extensions $\gamma = \beta[y{\mapsto}\mathsf{c}]$ for all $\beta = [x{\mapsto}\mathsf{a}, z{\mapsto}\mathsf{b}]$. The main difference is that $\mathcal{A}$ is not expected to give us a concrete semantic object that embodies the strategy, while $\mathcal{B}$ needs to explicitly provide such an object, namely $f^{\mathcal{B}}$.

Viewing this picture of φ through the lens of separateness, we observe that the variables x and y do not co-occur in any atom in φ and neither do x and z. Moreover, we noted on page 19 that there is the equivalent sentence

$$\begin{aligned}\varphi' := {} & \big(\big(\exists x_1.\, P(x_1) \wedge Q(x_1)\big) \to \forall z_1\exists y_1.\, R(y_1,z_1)\big)\\ & \wedge \big(\big(\exists x_2.\, P(x_2) \wedge \neg Q(x_2)\big) \to \forall z_2\exists y_2.\, \neg R(y_2,z_2)\big)\\ & \wedge \big(\big(\exists x_3.\, \neg P(x_3) \wedge Q(x_3)\big) \to \forall z_3\exists y_3.\, \neg R(y_3,z_3)\big)\\ & \wedge \big(\big(\exists x_4.\, \neg P(x_4) \wedge \neg Q(x_4)\big) \to \forall z_4\exists y_4.\, R(y_4,z_4)\big)\end{aligned}$$

in which each existential quantifier occurs in the scope of at most one universal quantifier. This raises the question of whether y in the original φ really depends on two universally quantified variables and, if so, how strong this dependence is. The short answer is: yes, y in φ depends on x and z. However, while the dependence of y on z could be considered of the "usual" kind, the dependence of y on x is of a weaker form. To verbalize this distinction, we shall call the former kind of dependence *strong* and the latter kind *weak*, as already proposed in Chapter 2.

Definition 4.0.1 (Weak dependence)**.** *Recall that we tacitly assume that distinct quantifiers in formulas bind distinct variables and that no variable has free and bound occurrences.*

Consider any satisfiable relational first-order sentence ψ in negation normal form that contains some subformula $\chi := \exists y.\, \chi'(\bar{\mathrm{u}}, \bar{\mathrm{v}}, \bar{\mathrm{x}}, y)$ such that the variables from $\bar{\mathrm{u}}$ and $\bar{\mathrm{x}}$ are universally quantified in ψ, and the variables from $\bar{\mathrm{v}}$ are existentially quantified in ψ. Let ψ_{Sk} be the result of replacing every occurrence of y in ψ with the Skolem term $f(\bar{\mathrm{u}}, \bar{\mathrm{x}})$ for some fresh Skolem function f. Then, y depends weakly *on the variables in $\bar{\mathrm{x}}$, if every model $\mathcal{A} \models \psi_{\mathrm{Sk}}$ can be turned into a model $\mathcal{B} \models \psi_{\mathrm{Sk}}$ by replacing $f^{\mathcal{A}}$ with some mapping $f^{\mathcal{B}}$ that satisfies the following property. There exists a* finite *family of mappings $\big(g_i : \mathsf{A}^{|\bar{\mathrm{u}}|} \to \mathsf{A}\big)_{i \in I}$ and some mapping $h : \mathsf{A}^{|\bar{\mathrm{x}}|} \to I$ such that we have $f^{\mathcal{B}}(\bar{\mathsf{a}}, \bar{\mathsf{b}}) := g_{h(\bar{\mathsf{b}})}(\bar{\mathsf{a}})$ for all $\bar{\mathsf{a}} \in \mathsf{A}^{|\bar{\mathrm{u}}|}$ and $\bar{\mathsf{b}} \in \mathsf{A}^{|\bar{\mathrm{x}}|}$.*

Let us go back to the exemplary sentence $\varphi = \forall xz \exists y.\, P(x) \leftrightarrow \big(Q(x) \leftrightarrow R(y,z)\big)$ and the insight the every model of $\mathcal{A}$ is equipped with some strategy $\sigma : \mathsf{A} \times \mathsf{A} \to \mathsf{A}$ that, given any two values a and b for x and z, respectively, provides a suitable value for y such that $\mathcal{A} \models P(\mathsf{a}) \leftrightarrow \big(Q(\mathsf{a}) \leftrightarrow R\big(\sigma(\mathsf{a},\mathsf{b}),\mathsf{b}\big)\big)$. The syntactic structure of the quantifier-free part of φ is such that we can group all the domain elements a that can be assigned to x into four categories: Category I satisfies $P(\mathsf{a}) \wedge Q(\mathsf{a})$, Category II satisfies $P(\mathsf{a}) \wedge \neg Q(\mathsf{a})$, Category III satisfies $\neg P(\mathsf{a}) \wedge Q(\mathsf{a})$, and Category IV satisfies $\neg P(\mathsf{a}) \wedge \neg Q(\mathsf{a})$. In order to make the sentence true under $\mathcal{A}$, the strategy σ does not have to distinguish different elements a, a' that belong to the same category. Let $\mathcal{S} := \big\{\{P(x), Q(x)\}, \{P(x), \neg Q(x)\}, \{\neg P(x), Q(x)\}, \{\neg P(x), \neg Q(x)\}\big\}$ be a collection of sets representing the Categories I to IV. We shall call such sets *fingerprints*. From the perspective of σ, any input individual a in σ's first argument is sufficiently characterized by its fingerprint. Therefore, we can find a finite family of component strategies $\big(\tau_S\big)_{S \in \mathcal{S}}$, containing one strategy $\tau_S : \mathsf{A} \to \mathsf{A}$ for each of the Categories I to IV, such that the combined strategy $\tau : \mathsf{A} \times \mathsf{A} \to \mathsf{A}$ — given by $\tau(\mathsf{a},\mathsf{b}) := \tau_S(\mathsf{b})$ whenever a's fingerprint is S — also yields $\mathcal{A} \models P(\mathsf{a}) \leftrightarrow \big(Q(\mathsf{a}) \leftrightarrow R\big(\tau(\mathsf{a},\mathsf{b}),\mathsf{b}\big)\big)$. This point of view highlights the fact that a good choice for y's value does not depend on the individual a assigned to x but rather on a's fingerprint S_{a}. Hence, the strategy witnesses the fact that y depends only weakly on x.

Although Definition 4.0.1 suggests a more general notion of weak dependence, we shall focus on the kind of weak dependence that can be phrased in terms of fingerprints. An important property of the different sorts of fingerprints we consider in the present chapter is that there are only finitely many fingerprints available at a time. Our concept of fingerprints shall always be based on the syntax of the formula at hand and the number of fingerprints is always bounded by some function in the length of the formula or a similar quantity. If we speak of weak dependences in a given formula φ, then it is understood that they are weak with respect to *all models* of φ. We will encounter strong dependences, however, that behave similar to weak dependences in certain models. In particular, classes of sentences with strong dependences may still enjoy the *finite model property*, i.e. any satisfiable sentence in the class possesses a finite model. This is, for instance, the case for GAF. We shall occasionally refer to the dependences in such classes as *finitely controllable*[1].

Let us, once more, go back to the above example. As the case distinction $\tau(\mathsf{a},\mathsf{b}) := \tau_S(\mathsf{b})$, where S denotes a's fingerprint, ranges over a finite number of fingerprints $S \in \mathcal{S}$ that is bounded irrespectively of $\mathcal{A}$, we can also express the idea entirely at the syntactic level using second-order quantifiers: $\exists g_1 \ldots g_4 . \forall xz . \bigvee_{i=1}^{4} P(x) \leftrightarrow \big(Q(x) \leftrightarrow R(g_i(z), z)\big)$. This sentence can be read as the result of applying to φ a non-standard form of Skolemization that is sensitive to the difference between weak and strong dependences. To this end, we introduce more than one Skolem term for the single quantifier $\exists y$, each of which has only the variable z as argument. In the field of proof complexity it is known that using different forms of Skolemization can have dramatic effects on the length of shortest refutation proofs [BL94, Egl94]. Hence, the proposed form of Skolemization might be an interesting object of study in that context. In Section 7.2, we will briefly look into the topic.

In the rest of the chapter, we shall concentrate on the description of syntactic criteria that entail weakness of dependences in the context of SF, GBSR, and GAF. Indeed we will observe

[1] This notion is also used in the literature, e.g. in [DG79], to refer to classes of first-order sentences that enjoy the finite model property.

that in GBSR sentences all occurring dependences between existentially and universally quantified variables are weak. In GAF sentences, on the other hand, also strong dependences might occur, namely between existentially quantified variables and their respective reference variable.

The main tools for studying dependences in SF, GBSR, and GAF sentences will be developed in the framework of *model-checking games* in the spirit of Hintikka [Hin73] and Henkin [Hen61]. Roughly speaking, Hintikka ([Hin73], Section III.8) defines such games as follows. We a priori fix a prenex sentence, e.g. φ from above, and some structure $\mathcal{A}$ and play against some opponent, "some recalcitrant *malin génie* making the most of his chances of frustrating us."([Hin73], page 63) Our goal in a play is to create an assignment for the variables in φ that makes the quantifier-free part of φ true under $\mathcal{A}$. The values for the variables are successively chosen from left to right in the order the respective quantifiers appear in φ's quantifier prefix. Our opponent picks the values for the universally quantified variables, we have control over the existentially quantified ones. It is easy to see that we have a winning strategy (also: satisfying strategy) for this game if and only if φ is satisfied under $\mathcal{A}$ in the standard semantics. In the presence of weak dependences, there are winning strategies that consider the fingerprints of the elements chosen by our opponent rather than the particular individuals. Individuals have to be considered only where strong dependences occur. We shall call such special kinds of winning strategies *(semi-)uniform*, cf. Definitions 4.2.6 and 4.3.3. They respond uniformly to moves by our opponent that result in identical fingerprints.

4.1 The Simple Case of SF

As a starter, we investigate the dependence patterns that emerge in SF sentences. We will do this on a rather informal level to get acquainted with the basic ideas. The formal treatment for GBSR in the subsequent section will of course subsume the case of SF.

As it turns out, all dependences of existential on universal variables in any satisfiable SF sentence φ are weak. By Definition 4.0.1, it follows that the sentence φ_{Sk}, which is the result of exhaustively Skolemizing all existentially quantified variables in φ, has a model $\mathcal{A}$ that interprets all Skolem functions f in φ_{Sk} with some function $f^{\mathcal{A}}$ whose image $\left\{ f^{\mathcal{A}}(\bar{\mathsf{a}}) \;\middle|\; \bar{\mathsf{a}} \in \mathsf{A}^{\mathrm{arity}(f)} \right\}$ is finite. This has an interesting consequence: any model $\mathcal{A} \models \varphi$ contains some finite substructure $\mathcal{B}$ that is also a model of φ. This certainly already follows from our earlier observation that there is some BSR sentence equivalent to φ. But this time, the argument emphasizes a semantic point of view and the original sentence need not be transformed syntactically.

Next, we outline the reasoning that leads to the above observation. Consider any satisfiable SF sentence $\varphi = \forall x_1 \exists y_1 \ldots \forall x_n \exists y_n . \psi$ where ψ is quantifier free and let $\mathcal{A}$ be any model of φ. Then, the two sets $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_n\}$ are separated in φ. In order to show that there is a finite substructure of $\mathcal{A}$ that still satisfies φ, we consider the model-checking game associated with the pair $\varphi, \mathcal{A}$. Recall that we play against some opponent: we take turns while constructing a variable assignment for the variables in φ's quantifier prefix. In the i-th round, our opponent picks the same value for x_i from A and right afterwards we are to choose a value for y_i from A. The play ends, as soon as all variables are assigned a value. If the resulting variable assignment is satisfying for the matrix ψ under $\mathcal{A}$, we win. Otherwise, our opponent wins. Evidently, since φ is satisfied by $\mathcal{A}$, there must exist a strategy that guarantees our victory, if we adhere to it.

The proof of the existence of the satisfying finite substructure $\mathcal{B}$ of $\mathcal{A}$ rests on two aspects:

(1) There exists a mapping μ that labels tuples of domain elements of $\mathcal{A}$ with suitable *fingerprints* taken from a finite supply.

(2) There exists a *winning strategy* σ for us that is *uniform* in the following sense. Consider the i-th move, in which our opponent has already assigned values to $x_1, \ldots, x_i$. Based on these values, the strategy σ now proposes a value for y_i. σ is uniform if for all sequences $\mathsf{a}_1, \ldots, \mathsf{a}_i$ and $\mathsf{b}_1, \ldots, \mathsf{b}_i$ of domain elements that could have been assigned to $x_1, \ldots, x_i$, we

have $\sigma(\mathsf{a}_1, \ldots, \mathsf{a}_i) = \sigma(\mathsf{b}_1, \ldots, \mathsf{b}_i)$, whenever

$$\begin{aligned} \mu(\mathsf{a}_1, \ldots, \mathsf{a}_{i-1}, \mathsf{a}_i) &= \mu(\mathsf{b}_1, \ldots, \mathsf{b}_{i-1}, \mathsf{b}_i) \ , \\ \mu(\mathsf{a}_1, \ldots, \mathsf{a}_{i-1}) &= \mu(\mathsf{b}_1, \ldots, \mathsf{b}_{i-1}) \ , \\ &\vdots \\ \mu(\mathsf{a}_1, \mathsf{a}_2) &= \mu(\mathsf{b}_1, \mathsf{b}_2) \ , \text{ and} \\ \mu(\mathsf{a}_1) &= \mu(\mathsf{b}_1) \ . \end{aligned}$$

In other words, the strategy σ is uniform if it proposes the same response for all inputs whose prefixes are labeled identically by μ. As μ can assign only finitely many labels to sequences of domain elements, this means that a uniform winning strategy σ gets along with only finitely many different moves, even if the universal player has infinitely many moving options.

From this point on the argument for the existence of a finite substructure of $\mathcal{A}$ that is still satisfying for φ roughly proceeds as follows. If we Skolemize φ exhaustively, and thus replace every occurrence of an existentially quantified variable by an appropriate Skolem term, the strategy σ induces an interpretation for the introduced Skolem functions. As σ is uniform, the image of each of these functions is finite. The Skolem functions then generate only a finite subset of $\mathcal{A}$'s domain, and thus, by the Substructure Lemma, induce a finite substructure of $\mathcal{A}$ that is still satisfying for φ.

It remains to discuss the mapping μ and the available fingerprints. Let At_x be the set of atoms in φ which contain at least one variable from the list $x_1, \ldots, x_n$. Then, At_x comprises exactly the atoms in φ that are affected by the values assigned to $x_1, \ldots, x_n$. We define μ recursively as follows:

Base case: for every sequence $\mathsf{a}_1, \ldots, \mathsf{a}_n$ of domain elements we set
$$\mu(\mathsf{a}_1, \ldots, \mathsf{a}_n) := \big\{ A(x_1, \ldots, x_n) \in \mathrm{At}_x \bigm| \mathcal{A} \models A(\mathsf{a}_1, \ldots, \mathsf{a}_n) \big\}.$$

Inductive case: for any sequence $\mathsf{a}_1, \ldots, \mathsf{a}_i$ of domain elements we set
$$\mu(\mathsf{a}_1, \ldots, \mathsf{a}_i) := \big\{ S \bigm| S = \mu(\mathsf{a}_1, \ldots, \mathsf{a}_i, \mathsf{b}) \text{ for some domain element } \mathsf{b} \big\}.$$

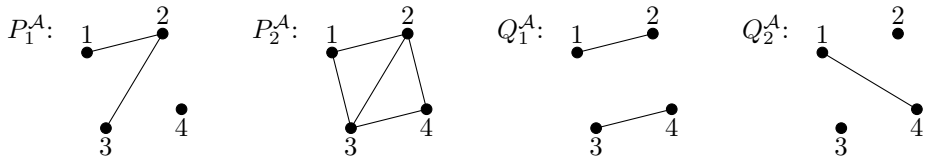
Hence, μ ranges over iterated power sets of the set At_x; the nesting becomes deeper the fewer arguments μ gets as input, i.e. $\mu(\mathsf{a}_1, \ldots, \mathsf{a}_i) \in \mathcal{P}^{n-i+1}\mathrm{At}_x$.

From the perspective of the argument sketched above, the i-th move a uniform strategy σ proposes based on μ is determined by the set of atoms that are factually or potentially satisfied by the domain elements our opponent has chosen for $x_1, ..., x_i$ and is potentially going to choose for $x_{i+1}, \ldots, x_n$. It is one peculiarity of SF that this set of atoms is not affected by the choices made for $y_1, \ldots, y_n$. This leads to the existence of uniform winning strategies whenever the SF formula at hand is satisfiable.

Example 4.1.1. *Consider the SF sentence*

$$\varphi := \forall x_1 \exists y_1 \forall x_2 \exists y_2 . \big(P_1(x_1, x_2) \leftrightarrow Q_1(y_1, y_2)\big) \wedge \big(P_2(x_1, x_2) \leftrightarrow Q_2(y_1, y_2)\big) \ .$$

Let $\mathcal{A}$ be the structure with domain $\mathsf{A} := \{1, 2, 3, 4\}$ that interprets P_1, P_2, Q_1, Q_2 by symmetric relations depicted below, which we, in addition, assume to be reflexive without depicting it.



Based on $\mathcal{A}$, we get the following fingerprint function μ. The pairs $\langle 1, 2\rangle$ and $\langle 2, 3\rangle$ are assigned the fingerprint $\big\{P_1(x_1, x_2), P_2(x_1, x_2)\big\}$ by μ. The same fingerprint is assigned to their symmetric

counterparts $\langle 2,1\rangle, \langle 3.2,\rangle$ *and to the reflexive pairs* $\langle 1,1\rangle$, $\langle 2,2\rangle$, $\langle 3,3\rangle$, *and* $\langle 4,4\rangle$. *The pairs* $\langle 1,3\rangle, \langle 2,4\rangle, \langle 3,4\rangle$ *and their symmetric counterparts are assigned the fingerprint* $\{P_2(x_1,x_2)\}$ *by* μ. *Finally, the pairs* $\langle 1,4\rangle$ *and* $\langle 4,1\rangle$ *have the fingerprint* $\emptyset$. *Concerning the single elements, we get the fingerprints*

$$\mu(1) = \mu(4) = \big\{\{P_1(x_1,x_2), P_2(x_1,x_2)\}, \{P_2(x_1,x_2)\}, \emptyset\big\} \text{ , and}$$
$$\mu(2) = \mu(3) = \big\{\{P_1(x_1,x_2), P_2(x_1,x_2)\}, \{P_2(x_1,x_2)\}\big\} \text{ .}$$

Having all this, a satisfying strategy that is uniform with respect to μ *is given, for instance, by setting* $\sigma(\mathsf{a}) := 4$ *and*

$$\sigma(\mathsf{a},\mathsf{b}) := \begin{cases} 4 & \text{if } \mu(\mathsf{a},\mathsf{b}) = \{P_1(x_1,x_2), P_2(x_1,x_2)\}, \\ 3 & \text{if } \mu(\mathsf{a},\mathsf{b}) = \{P_1(x_1,x_2)\}, \\ 1 & \text{if } \mu(\mathsf{a},\mathsf{b}) = \{P_2(x_1,x_2)\}, \text{ and} \\ 2 & \text{if } \mu(\mathsf{a},\mathsf{b}) = \emptyset \end{cases}$$

for all $\mathsf{a},\mathsf{b} \in \mathsf{A}$. *Then, we get*

$$\mathcal{A} \models \big(P_1(\mathsf{a},\mathsf{b}) \leftrightarrow Q_1(\sigma(\mathsf{a}),\sigma(\mathsf{a},\mathsf{b}))\big) \wedge \big(P_2(\mathsf{a},\mathsf{b}) \leftrightarrow Q_2(\sigma(\mathsf{a}),\sigma(\mathsf{a},\mathsf{b}))\big)$$

for any choice of $\mathsf{a},\mathsf{b} \in \mathsf{A}$, *which entails that* σ *is indeed a winning strategy.*

Remark 4.1.2. *In his thesis [Her30], Section 9 of Chapter 2, Herbrand gave a proof of the decidability of MFO using an approach which differs from the ones published by then by Löwenheim [Löw15], Skolem [Sko19], Behmann [Beh22], and Hilbert and Ackermann [HA28] (page 77). Herbrand's proof uses an equivalence-preserving transformation of MFO sentences into a normal form in which no quantifier occurs in the scope of another quantifier. This part is similar to the approaches we took in Chapter 3 to devise translations from our novel extended fragments into the respective original fragment.*

Starting from this normal form and the following arguments, Quine [Qui69] extrapolated his proof of the decidability of the class of homogeneous k-adic sentences, *i.e. the class of FL sentences in which all occurring predicate symbols have arity* k. *In retrospect, it seems fair to say that Quine was on a track that eventually could have lead to the discovery of SF and to a proof of its decidability. Quine's arguments for the decidability of homogeneous* k*-adic sentences are very closely linked to the concept of fingerprints. For example, given a* homogeneous dyadic sentence φ φ *containing the atoms* $P(x,y), Q(x,y), R(x,y)$, *Quine defines* super-constituents *to be sentences of the form* $\exists x.\ \bigwedge_i [\neg]\exists y.\, [\neg]P(x,y) \wedge [\neg]Q(x,y) \wedge [\neg]R(x,y)$, *where* $[\neg]$ *means that the negation sign may be present or not and where every conjunct differs from every other conjunct in the presence or absence of at least one negation sign within the scope of* $\exists y$. *It is easy to see that such a sentence corresponds to a fingerprint. For instance, the super-constituent*

$$\begin{aligned} \exists x.\quad & \neg\exists y.\big(\ P(x,y) \wedge\ Q(x,y) \wedge\ R(x,y)\big) \\ \wedge\ & \exists y.\big(\ P(x,y) \wedge\ Q(x,y) \wedge \neg R(x,y)\big) \\ \wedge\ & \exists y.\big(\ P(x,y) \wedge \neg Q(x,y) \wedge\ R(x,y)\big) \\ \wedge\ & \neg\exists y.\big(\ P(x,y) \wedge \neg Q(x,y) \wedge \neg R(x,y)\big) \\ \wedge\ & \neg\exists y.\big(\neg P(x,y) \wedge\ Q(x,y) \wedge\ R(x,y)\big) \\ \wedge\ & \exists y.\big(\neg P(x,y) \wedge\ Q(x,y) \wedge \neg R(x,y)\big) \\ \wedge\ & \neg\exists y.\big(\neg P(x,y) \wedge \neg Q(x,y) \wedge\ R(x,y)\big) \\ \wedge\ & \neg\exists y.\big(\neg P(x,y) \wedge \neg Q(x,y) \wedge \neg R(x,y)\big) \end{aligned}$$

corresponds to the fingerprint $\big\{\{P(x,y), Q(x,y)\}, \{P(x,y), R(x,y)\}, \{Q(x,y)\}\big\}$. *One can now argue that every sentence* $\bigwedge_j [\neg]C_i$, *where* C_i *ranges over all super-constituents with respect to* φ*'s vocabulary, induces a finite structure and that it is sufficient to consider only such structures in order to find a model for* φ.

We shall use fingerprints in a different way, though, when constructing finite models from uniform winning strategies. Nevertheless, in the light of the above said, it might have been pure coincidence that Quine has extrapolated Herbrand's ideas in the direction of the fluted fragment rather than the direction of the separated fragment. In Section 7.1 we will establish a formal link between fingerprints and sentences that are similar to the generalization of super-constituents for k-adic sentences.[2]

4.2 GBSR Sentences and the Existence of Uniform Winning Strategies

It is obvious that all dependences in any satisfiable BSR sentences $\varphi := \exists\bar{\mathrm{y}}\forall\bar{\mathrm{x}}.\,\psi$ are trivially weak: existential quantifiers never occur within the scope of universal quantifiers. Hence, each $y \in \bar{\mathrm{y}}$ is independent of any $x \in \bar{\mathrm{x}}$, and for any model $\mathcal{A}$ we need at most a supply of $|\bar{\mathrm{y}}|$ domain elements from which suitable values for the $y \in \bar{\mathrm{y}}$ can be picked. Concerning expressiveness, BSR sentences are in the following sense prototypical for the class of sentences with only weak dependences.

Theorem 4.2.1. *Consider any satisfiable relational sentence φ in which all dependences of existentially quantified variables on universally quantified variables are weak. Let φ_{Sk} be the sentence that results from φ by exhaustive Skolemization of all existentially quantified variables. Suppose Ω is the set containing exactly all the Skolem functions introduced in φ_{Sk}. Then, we observe the following.*

(i) There is some positive integer m such that for every model $\mathcal{A} \models \varphi_{\mathrm{Sk}}$ there is some model $\mathcal{B} \models \varphi_{\mathrm{Sk}}$ that differs from $\mathcal{A}$ only in the interpretation of the Skolem functions in Ω and for which the set $\bigcup_{f\in\Omega}\{f^{\mathcal{B}}(\bar{\mathsf{a}}) \mid \bar{\mathsf{a}} \in \mathsf{B}^{\mathrm{arity}(f)}\}$ contains at most m domain elements.

(ii) There is some BSR sentence φ' that is equivalent to φ.

Proof sketch.

Ad (i). Suppose that there is no such integer m. For every positive n let $z_1, \ldots, z_n$ be fresh variables that do not occur in φ and let φ'_n be the sentence that results from φ by iteratively replacing every subformula $\exists y.\,\psi$ with $\bigvee_{1\leq i\leq n}\psi[y/z_i]$. Due to our assumptions — φ is satisfiable and m does not exist —, for every positive integer n the sentence $\varphi_{\mathrm{Sk}} \wedge \neg\exists z_1 \ldots z_n.\,\varphi'_n$ is satisfiable. But then, by compactness of first-order logic, the set of sentences $\{\varphi_{\mathrm{Sk}}\} \cup \{\neg\exists z_1 \ldots z_n.\,\varphi'_n \mid n \geq 1\}$ is satisfied by some model $\mathcal{A}$. Moreover, there is no model $\mathcal{B} \models \varphi_{\mathrm{Sk}}$ that differs from $\mathcal{A}$ only in the interpretation of the Skolem functions $f \in \Omega$ and for which the set $\bigcup_{f\in\Omega}\{f^{\mathcal{B}}(\bar{\mathsf{a}}) \mid \bar{\mathsf{a}} \in \mathsf{B}^{\mathrm{arity}(f)}\}$ is finite. This contradicts our assumption that all dependences in φ are weak, as the latter entails that every model $\mathcal{A} \models \varphi_{\mathrm{Sk}}$ can be turned into some model $\mathcal{B} \models \varphi_{\mathrm{Sk}}$ by replacing every $f^{\mathcal{A}}$ with some mapping $f^{\mathcal{B}}$ with a finite image. ◊

Ad (ii). Let m be the integer whose existence is stipulated in (i). Then, under any of φ's models each existential quantifier $\exists y$ in φ needs to range over at most m domain elements. Hence, we can replace any subformula χ of the form $\exists y.\,\psi(y, \bar{\mathrm{v}})$ in φ with a finite disjunction $\bigvee_{i=1}^{m}\psi(z_i, \bar{\mathrm{v}})$, where the z_i are fresh variables, and add the quantifier block $\exists z_1 \ldots z_m$ to the front. The resulting sentence $\exists z_1 \ldots z_m.\,\varphi\big[\chi \,/\, \bigvee_{i=1}^{m}\psi(z_i, \bar{\mathrm{v}})\big]$ is equivalent to φ. Applying this transformation exhaustively eventually leads to an equivalent BSR sentence. □

As we already know from Theorem 3.2.7, the sentence φ Theorem 4.2.1 may be non-elementarily more succinct than any equivalent BSR sentence, as the numbers m in part (i) of the theorem may be very large.

In contrast to BSR sentences, GBSR sentences may contain non-trivial weak dependences. We aim to show in the present section that GBSR sentences may, on the other hand, not contain

[2] See also Remark 7.1.4 on page 187 for a brief discussion of the connection between fingerprints and the model-theoretic concept of *types*.

any strong dependences. We have already obtained indirect evidence that all dependences in a GBSR sentence are weak in Section 3.5, when we devised the equivalence-preserving translation from GBSR into BSR. However, this does not reveal much about the underlying semantic reasons. A direct approach in the framework of model-checking games sheds more light on the involved semantic properties and concepts, i.e. uniform satisfying strategies based on fingerprints.

The following are key concepts: *fingerprints* are, again, sets of sets of ... sets of atoms that characterize certain classes of indistinguishable tuples of domain elements by finite means; *fingerprint functions* $\mu_{\ell,k}$ assign fingerprints to such tuples; *μ-uniform strategies* select domain elements for existentially quantified variables exclusively depending on the fingerprints of the values assigned to preceding universally quantified variables. We base our considerations on some fixed GBSR sentence $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \psi$ in standard form. Let the sets At, $\bar{\mathrm{x}}$, and $\bar{\mathrm{y}}$ be defined like in Definition 3.4.1. Then, At can be partitioned into (possibly empty) sets $\mathrm{At}_0, \ldots, \mathrm{At}_n$ such that Conditions (i) and (ii) of Definition 3.4.1 are met. Let $\mathcal{A}$ be any structure over the vocabulary of φ and consider the model-checking game associated with φ over $\mathcal{A}$. We next define the notion of *strategy* in the context of GBSR.

φ

$\mathcal{A}$

Definition 4.2.2 (Strategy, satisfying strategy, outcome)**.** *A* strategy σ *comprises a tuple of* n *mappings* $\langle \sigma_1, \ldots, \sigma_n \rangle$ *with signatures* $\sigma_i : \mathsf{A}^{|\bar{\mathrm{x}}_1|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{x}}_i|} \to \mathsf{A}^{|\bar{\mathrm{y}}_i|}$*. A strategy* σ *is* satisfying for φ (under $\mathcal{A}$) *if*

$$\mathcal{A}, [\bar{\mathrm{x}}_1 \mapsto \bar{\mathsf{a}}_1, \ldots, \bar{\mathrm{x}}_n \mapsto \bar{\mathsf{a}}_n, \bar{\mathrm{y}}_1 \mapsto \sigma_1(\bar{\mathsf{a}}_1), \ldots, \bar{\mathrm{y}}_n \mapsto \sigma_n(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n)] \models \psi$$

holds for every choice of tuples $\bar{\mathsf{a}}_1 \in \mathsf{A}^{|\bar{\mathrm{x}}_1|}, \ldots, \bar{\mathsf{a}}_n \in \mathsf{A}^{|\bar{\mathrm{x}}_n|}$.

For all $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n$ *with* $\bar{\mathsf{a}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}_i|}$ *we denote by* $\mathrm{out}_\sigma(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n)$ *the set*

$\mathrm{out}_\sigma(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n)$

$$\left\{ A(\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n) \in \mathrm{At} \;\middle|\; \mathcal{A} \models A\big(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n, \sigma_1(\bar{\mathsf{a}}_1), \ldots, \sigma_n(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n)\big) \right\} ,$$

called the outcome of $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n$ under σ. *By* Out_σ *we denote the* set of all possible outcomes under σ, *i.e.* $\mathrm{Out}_\sigma := \left\{ \mathrm{out}_\sigma(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n) \;\middle|\; \bar{\mathsf{a}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}_i|} \right\}$.

Out_σ

Satisfying strategies can be considered *winning strategies* against our opponent in the model-checking game associated with φ over $\mathcal{A}$. If we adhere to a satisfying strategy σ during a play, then every possible outcome in Out_σ represents a satisfying assignment of truth values to the atoms in ψ — an atom A is *true* if and only if is belongs to the outcome. Hence, the structure $\mathcal{A}$ satisfies φ if and only if there is a satisfying strategy for φ.

Every strategy σ induces a structure $\mathcal{A}|_\sigma$ that is given by the substructure of $\mathcal{A}$ with the domain

$\mathcal{A}|_\sigma$

$$\mathsf{A}|_\sigma := \left\{ \mathsf{a} \in \mathsf{A} \;\middle|\; \sigma_k(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_k) = \langle \ldots, \mathsf{a}, \ldots \rangle \text{ for some } \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_k \right\} .$$

As we assume φ to be relational, such a substructure $\mathcal{A}|_\sigma$ exists.

Lemma 4.2.3. *If a strategy* σ *is satisfying for* φ *under* $\mathcal{A}$*, then* σ *is satisfying for* φ *under* $\mathcal{A}|_\sigma$*.*

Proof. It is easy to show by induction on the structure of φ that for every variable assignment β over $\mathcal{A}|_\sigma$'s domain $\mathcal{A}, \beta \models \varphi$ implies $\mathcal{A}|_\sigma, \beta \models \varphi$. Since σ is satisfying for φ under $\mathcal{A}$, we have

$$\mathcal{A}, [\bar{\mathrm{x}}_1 \mapsto \bar{\mathsf{a}}_1, \ldots, \bar{\mathrm{x}}_n \mapsto \bar{\mathsf{a}}_n, \bar{\mathrm{y}}_1 \mapsto \sigma_1(\bar{\mathsf{a}}_1), \ldots, \bar{\mathrm{y}}_n \mapsto \sigma_n(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n)] \models \varphi$$

for any choice of tuples $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n$ with $\bar{\mathsf{a}}_i \in (\mathsf{A}|_\sigma)^{|\bar{\mathrm{x}}_i|} \subseteq \mathsf{A}^{|\bar{\mathrm{x}}_i|}$. Hence, we get

$$\mathcal{A}|_\sigma, [\bar{\mathrm{x}}_1 \mapsto \bar{\mathsf{a}}_1, \ldots, \bar{\mathrm{x}}_n \mapsto \bar{\mathsf{a}}_n, \bar{\mathrm{y}}_1 \mapsto \sigma_1(\bar{\mathsf{a}}_1), \ldots, \bar{\mathrm{y}}_n \mapsto \sigma_n(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n)] \models \varphi$$

for any choice of $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n$ with $\bar{\mathsf{a}}_i \in (\mathsf{A}|_\sigma)^{|\bar{\mathrm{x}}_i|}$. In other words, σ is satisfying for φ under $\mathcal{A}|_\sigma$. □

If φ is satisfied by $\mathcal{A}$, and if all dependences in φ are weak, then there must be a special form of satisfying strategies whose image only covers a finite portion of $\mathcal{A}$'s domain. Such a strategy σ induces a finite substructure $\mathcal{A}|_\sigma$ of $\mathcal{A}$ that also satisfies φ. In order to find σ, we need to identify the key features of domain elements that make them distinguishable by the formula φ. We express these features by suitable *fingerprints*.

Definition 4.2.4 (Fingerprint functions $\mu_{\ell,k}$)**.** *We define the family of* fingerprint functions $\mu_{\ell,k}$ *with* $0 \leq \ell < k \leq n$ *as follows:* $\mu_{\ell,k}$

$\mu_{\ell,n} : \mathsf{A}^{|\bar{\mathrm{y}}_1|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{y}}_\ell|} \times \mathsf{A}^{|\bar{\mathrm{x}}_{\ell+1}|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{x}}_n|} \to \mathcal{P}\mathrm{At}_\ell$ *such that for all tuples* $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_n$ *and every atom* $A(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_\ell, \bar{\mathrm{x}}_{\ell+1}, \ldots, \bar{\mathrm{x}}_n) \in \mathrm{At}_\ell$ *we have* $A \in \mu_{\ell,n}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_n)$ *if and only if* $\mathcal{A} \models A(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_n)$*;*

$\mu_{\ell,n-1} : \mathsf{A}^{|\bar{\mathrm{y}}_1|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{y}}_\ell|} \times \mathsf{A}^{|\bar{\mathrm{x}}_{\ell+1}|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{x}}_{n-1}|} \to \mathcal{P}^2\mathrm{At}_\ell$ *such that for all* $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_{n-1}$ *and every* $S \in \mathcal{P}\mathrm{At}_\ell$ *we have* $S \in \mu_{\ell,n-1}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_{n-1})$ *if and only if there is some* $\bar{\mathsf{b}}_n$ *for which* $\mu_{\ell,n}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_{n-1}, \bar{\mathsf{b}}_n) = S$*;*

$\vdots$

$\mu_{\ell,\ell+1} : \mathsf{A}^{|\bar{\mathrm{y}}_1|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{y}}_\ell|} \times \mathsf{A}^{|\bar{\mathrm{x}}_{\ell+1}|} \to \mathcal{P}^{n-\ell}\mathrm{At}_\ell$ *such that for all tuples* $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}$ *and every* $S \in \mathcal{P}^{n-\ell-1}\mathrm{At}_\ell$ *we have* $S \in \mu_{\ell,\ell+1}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1})$ *if and only if there exists* $\bar{\mathsf{b}}_{\ell+2}$ *for which* $\mu_{\ell,\ell+2}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \bar{\mathsf{b}}_{\ell+2}) = S$*.*

$\mathrm{im}_\sigma(\mu_{\ell,k})$ *We denote the* image *of a fingerprint function* $\mu_{\ell,k}$ *under a strategy* $\sigma = \langle \sigma_1, \ldots, \sigma_n \rangle$ *by*

$$\mathrm{im}_\sigma(\mu_{\ell,k}) := \big\{ \mu_{\ell,k}\big(\sigma_1(\bar{\mathsf{b}}_1), \ldots, \sigma_\ell(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_\ell), \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k\big) \mid \bar{\mathsf{b}}_1 \in \mathsf{A}^{|\bar{\mathrm{x}}_1|}, \ldots, \bar{\mathsf{b}}_k \in \mathsf{A}^{|\bar{\mathrm{x}}_k|} \big\} .$$

Example 4.2.5. *Consider the sentence* $\varphi := \forall x_1 x_2 \exists y_1 \forall u \exists y_2 \forall v.\, R(x_1, u) \vee \big(P(x_2, v) \wedge T(y_1, y_2)\big)$*. We partition the set* $\mathrm{At} = \{R(x_1, u), P(x_2, v), T(y_1, y_2)\}$ *as follows:* $\mathrm{At}_0 := \{R(x_1, u), P(x_2, v)\}$*,* $\mathrm{At}_1 := \emptyset$*,* $\mathrm{At}_2 := \{T(y_1, y_2)\}$*,* $\mathrm{At}_3 := \emptyset$*. Regarding the images of the fingerprint functions* $\mu_{k,\ell}$*, we observe the following. Let* σ *be any strategy, based on any structure. Then, we have*
$\mathrm{im}_\sigma(\mu_{2,3}) \subseteq \big\{\{\}, \{T(y_1, y_2)\}\big\}$*,*
$\mathrm{im}_\sigma(\mu_{0,3}) \subseteq \big\{\{\}, \{R(x_1, u)\}, \{P(x_2, v)\}, \{R(x_1, u), P(x_2, v)\}\big\}$*,*
$\mathrm{im}_\sigma(\mu_{0,2}) \subseteq \Big\{\{\{\}\}, \big\{\{R(x_1, u)\}\big\}, \big\{\{P(x_2, v)\}\big\}, \big\{\{R(x_1, u), P(x_2, v)\}\big\},$
$\big\{\{\}, \{P(x_2, v)\}\big\}, \big\{\{R(x_1, u)\}, \{R(x_1, u), P(x_2, v)\}\big\}\Big\}.$
One aspect that restricts the image of $\mu_{0,2}$ *compared to the full set* $\mathcal{P}^2\mathrm{At}_2$ *is the fact that* u *and* v *do not co-occur in any atom. For every fingerprint* $S \in \mathrm{im}_\sigma(\mu_{0,2})$ *either* $R(x_1, u)$ *occurs in each and every set* $S' \in S$ *or in none, since the truth value of* $R(x_1, u)$ *does not depend on the value of* v*. Therefore, fingerprints such as*
$$\big\{\{\}, \{R(x_1, u), P(x_2, v)\}\big\} \text{ or } \big\{\{R(x_1, u)\}, \{P(x_2, v)\}, \{R(x_1, u), P(x_2, v)\}\big\}$$
cannot be the result of $\mu_{0,2}$*.*

Having a suitable notion of fingerprints at hand, we next define a special kind of strategies that highlight the weak nature of dependences in GBSR sentences and, hence, have a finite image.

Definition 4.2.6 (μ-uniformity)**.** *A strategy* $\sigma = \langle \sigma_1, \ldots, \sigma_n \rangle$ *is* μ-uniform *if for every* $k \leq n$ *the following holds. For all tuples* $\bar{\mathsf{b}}_1, \bar{\mathsf{b}}'_1 \in \mathsf{A}^{|\bar{\mathrm{x}}_1|}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{b}}'_k \in \mathsf{A}^{|\bar{\mathrm{x}}_k|}$ *we have* $\sigma_k(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_k) = \sigma_k(\bar{\mathsf{b}}'_1, \ldots, \bar{\mathsf{b}}'_k)$ *whenever for every* $k' \leq k$ *all of the following conditions are met:*

$$\begin{aligned}
\mu_{0,k'}\big(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k'}\big) &= \mu_{0,k'}\big(\bar{\mathsf{b}}'_1, \ldots, \bar{\mathsf{b}}'_{k'}\big) \ , \\
\mu_{1,k'}\big(\sigma_1(\bar{\mathsf{b}}_1), \bar{\mathsf{b}}_2, \ldots, \bar{\mathsf{b}}_{k'}\big) &= \mu_{1,k'}\big(\sigma_1(\bar{\mathsf{b}}'_1), \bar{\mathsf{b}}'_2, \ldots, \bar{\mathsf{b}}'_{k'}\big) \ , \\
&\vdots \\
\mu_{k'-1,k'}\big(\sigma_1(\bar{\mathsf{b}}_1), \ldots, \sigma_{k'-1}(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k'-1}), \bar{\mathsf{b}}_{k'}\big) &= \mu_{k'-1,k'}\big(\sigma_1(\bar{\mathsf{b}}'_1), \ldots, \sigma_{k'-1}(\bar{\mathsf{b}}'_1, \ldots, \bar{\mathsf{b}}'_{k'-1}), \bar{\mathsf{b}}'_{k'}\big) \ .
\end{aligned}$$

Intuitively, μ-uniformity of a strategy σ means that σ responds in the same way to inputs that have identical fingerprints. The next lemma provides the key argument to infer the existence of some satisfying μ-uniform strategy from the existence of any satisfying strategy.

Lemma 4.2.7. *For every strategy* $\sigma = \langle \sigma_1, \ldots, \sigma_n \rangle$ *there is a* μ*-uniform strategy* $\tau = \langle \tau_1, \ldots, \tau_n \rangle$ *such that* $\mathrm{Out}_\tau \subseteq \mathrm{Out}_\sigma$*.*

Proof. We start with two preliminary results.

Claim I: Let ℓ, k be two integers such that $0 \leq \ell < k \leq n$. Let $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k$ be tuples of domain elements, where $\bar{\mathsf{a}}_i \in \mathsf{A}^{|\bar{\mathrm{y}}_i|}$, $\bar{\mathsf{b}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}_i|}$ for every i. Let $\bar{\mathsf{c}}_{k+1}, \ldots, \bar{\mathsf{c}}_n$ and $\bar{\mathsf{d}}_{k+1}, \ldots, \bar{\mathsf{d}}_n$ be sequences of tuples with $\bar{\mathsf{c}}_i, \bar{\mathsf{d}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}_i|}$. Assume that $\bar{\mathsf{c}}_{k+1}, \ldots, \bar{\mathsf{c}}_n$ and $\bar{\mathsf{d}}_{k+1}, \ldots, \bar{\mathsf{d}}_n$ coincide in all positions that correspond to variables $x \in \bar{\mathrm{x}}$ occurring in At_ℓ. We have

$$\mu_{\ell,k'}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{c}}_{k+1}, \ldots, \bar{\mathsf{c}}_{k'}) = \mu_{\ell,k'}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{d}}_{k+1}, \ldots, \bar{\mathsf{d}}_{k'})$$

for every k' with $k \leq k' \leq n$.

Proof: We proceed inductively from $k' = n$ downwards.

Let $k' = n$. By definition of the sequences $\bar{\mathsf{c}}_{k+1}, \ldots, \bar{\mathsf{c}}_n$ and $\bar{\mathsf{d}}_{k+1}, \ldots, \bar{\mathsf{d}}_n$ we have

$$\mathcal{A} \models A(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{c}}_{k+1}, \ldots, \bar{\mathsf{c}}_n)$$

if and only if

$$\mathcal{A} \models A(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{d}}_{k+1}, \ldots, \bar{\mathsf{d}}_n)$$

for every atom $A(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_\ell, \bar{\mathrm{x}}_{\ell+1}, \ldots, \bar{\mathrm{x}}_n) \in \mathrm{At}_\ell$. Hence, we have

$$\mu_{\ell,n}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{c}}_{k+1}, \ldots, \bar{\mathsf{c}}_n) = \mu_{\ell,n}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{d}}_{k+1}, \ldots, \bar{\mathsf{d}}_n).$$

Let $k' < n$. Consider any set $S \in \mu_{\ell,k'}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{c}}_{k+1}, \ldots, \bar{\mathsf{c}}_{k'})$. By definition of $\mu_{\ell,k'}$, there must be some tuple $\bar{\mathsf{c}}_{k'+1}$ such that

$$S = \mu_{\ell,k'+1}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{c}}_{k+1}, \ldots, \bar{\mathsf{c}}_{k'}, \bar{\mathsf{c}}_{k'+1}).$$

By induction, $S = \mu_{\ell,k'+1}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{d}}_{k+1}, \ldots, \bar{\mathsf{d}}_{k'}, \bar{\mathsf{c}}_{k'+1})$ and thus we have $S \in \mu_{\ell,k'}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{d}}_{k+1}, \ldots, \bar{\mathsf{d}}_{k'})$.

Since this argument is symmetric, we obtain $\mu_{\ell,k'}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{c}}_{k+1}, \ldots, \bar{\mathsf{c}}_{k'}) = \mu_{\ell,k'}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{d}}_{k+1}, \ldots, \bar{\mathsf{d}}_{k'})$. ◊

Claim II: Let ℓ, k, k' be three integers such that $0 \leq \ell < k < k' \leq n$. Let $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k$ be tuples of domain elements, where $\bar{\mathsf{a}}_i \in \mathsf{A}^{|\bar{\mathrm{y}}_i|}$, $\bar{\mathsf{b}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}_i|}$ for every i. Consider two sequences of tuples $\bar{\mathsf{c}}_{k+1}, \ldots, \bar{\mathsf{c}}_{k'}$ and $\bar{\mathsf{d}}_{k+1}, \ldots, \bar{\mathsf{d}}_{k'}$ that coincide in all positions that correspond to variables x occurring in At_ℓ, where $\bar{\mathsf{c}}_i, \bar{\mathsf{d}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}_i|}$ for every i. We have

$$\mu_{\ell,k'}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{c}}_{k+1}, \ldots, \bar{\mathsf{c}}_{k'}) = \mu_{\ell,k'}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{d}}_{k+1}, \ldots, \bar{\mathsf{d}}_{k'}).$$

Proof: For $k' = n$ Claim II follows immediately from Claim I. For $k' < n$ we simply pad the sequences $\bar{\mathsf{c}}_{k+1}, \ldots, \bar{\mathsf{c}}_{k'}$ and $\bar{\mathsf{d}}_{k+1}, \ldots, \bar{\mathsf{d}}_{k'}$ with tuples $\bar{\mathsf{e}}_{k'+1}, \ldots, \bar{\mathsf{e}}_n$. Then, Claim II follows from Claim I applied to the sequences $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{c}}_{k+1}, \ldots, \bar{\mathsf{c}}_{k'}, \bar{\mathsf{e}}_{k'+1}, \ldots, \bar{\mathsf{e}}_n$ and $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{d}}_{k+1}, \ldots, \bar{\mathsf{d}}_{k'}, \bar{\mathsf{e}}_{k'+1}, \ldots, \bar{\mathsf{e}}_n$. ◊

For $i = 1, \ldots, n$ we define $\underline{\mathsf{A}}_i$ as abbreviation of $\mathsf{A}^{|\bar{\mathrm{x}}_1|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{x}}_i|}$. We construct certain representatives $\alpha_{k,\langle \overline{S}_0^{(k)}, \ldots, \overline{S}_{k-1}^{(k)} \rangle} \in \underline{\mathsf{A}}_k$ inductively as follows. The $\overline{S}_i^{(k)}$ stand for sequences $S_{i,i+1}^{(k)} \ldots S_{i,k}^{(k)}$ of fingerprints satisfying $S_{i,k}^{(k)} \in S_{i,k-1}^{(k)} \in \ldots \in S_{i,i+1}^{(k)}$.

$\underline{\mathsf{A}}_i$

$\alpha_{i,\langle \ldots \rangle}$, $\overline{S}_i^{(k)}$

Let $k = 1$. We partition $\underline{\mathsf{A}}_1$ into sets $\underline{\mathsf{A}}_{1,\langle S_0^{(1)} \rangle}$ with $S_0^{(1)} \in \mathrm{im}_\sigma(\mu_{0,1})$ by setting $\underline{\mathsf{A}}_{1,\langle S_0^{(1)} \rangle} := \big\{ \bar{\mathsf{b}}_1 \in \mathsf{A}^{|\bar{\mathrm{x}}_1|} \mid \mu_{0,1}(\bar{\mathsf{b}}_1) = S_0^{(1)} \big\}$. We pick one representative $\alpha_{1,\langle S_0^{(1)} \rangle} \in \underline{\mathsf{A}}_{1,\langle S_0^{(1)} \rangle}$ from every part.

$\underline{\mathsf{A}}_{i,\langle \ldots \rangle}$

Let $k > 1$. We construct subsets $\underline{\mathsf{A}}_{k,\langle \overline{S}_0^{(k)}, \ldots, \overline{S}_{k-1}^{(k)} \rangle} \subseteq \underline{\mathsf{A}}_k$ with $S_{0,j}^{(k)} \in \mathrm{im}_\sigma(\mu_{0,j}), \ldots, S_{k-1,j}^{(k)} \in \mathrm{im}_\sigma(\mu_{k-1,j})$ for every $j \leq k$ by setting $\underline{\mathsf{A}}_{k,\langle \overline{S}_0^{(k)}, \ldots, \overline{S}_{k-1}^{(k)} \rangle} :=$

$$\Big\{ \big\langle \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{b}}_k \big\rangle \;\Big|\; \bar{\mathsf{b}}_k \in \mathsf{A}^{|\bar{\mathrm{x}}_k|} \text{ and there is some } \alpha_{k-1,\langle \overline{S}_0^{(k-1)}, \ldots, \overline{S}_{k-2}^{(k-1)} \rangle} = \big\langle \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1} \big\rangle \text{ with } \bar{\mathsf{c}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}_i|} \text{ for every } i \text{ such that}$$

$$\mu_{0,k}\big(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{b}}_k\big) = S_0,$$

$$\mu_{1,k}\big(\sigma_1(\bar{\mathsf{c}}_1), \bar{\mathsf{c}}_2, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{b}}_k\big) = S_1,$$

$$\vdots$$

$$\mu_{k-2,k}\big(\sigma_1(\bar{\mathsf{c}}_1), \ldots, \sigma_{k-2}(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-2}), \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{b}}_k\big) = S_{k-2},$$

$$\begin{aligned}
&\mu_{k-1,k}\big(\sigma_1(\bar{\mathsf{c}}_1),\dots,\sigma_{k-1}(\bar{\mathsf{c}}_1,\dots,\bar{\mathsf{c}}_{k-1}),\bar{\mathsf{b}}_k\big) = S_{k-1},\\
&\overline{S}_0^{(k)} = \overline{S}_0^{(k-1)} S_0,\\
&\qquad\vdots\\
&\overline{S}_{k-2}^{(k)} = \overline{S}_{k-2}^{(k-1)} S_{k-2}, \text{ and}\\
&\overline{S}_{k-1}^{(k)} = S_{k-1}\big\}\ .
\end{aligned}$$

We pick one representative $\alpha_{k,\langle\overline{S}_0^{(k)},\dots,\overline{S}_{k-1}^{(k)}\rangle}$ from each nonempty $\underline{\mathsf{A}}_{k,\langle\overline{S}_0^{(k)},\dots,\overline{S}_{k-1}^{(k)}\rangle}$.

Having all the representatives $\alpha_{k,\langle\overline{S}_0^{(k)},\dots,\overline{S}_{k-1}^{(k)}\rangle}$ at hand, we inductively construct τ, starting from τ_1 and going to τ_n.

Let $k=1$. For every $\bar{\mathsf{b}}_1 \in \mathsf{A}^{|\bar{\mathrm{x}}_1|}$ we set $\tau_1(\bar{\mathsf{b}}_1) := \sigma_1(\alpha_{1,\langle S_0\rangle})$, where $S_0 := \mu_{0,1}\big(\bar{\mathsf{b}}_1\big)$.

Let $k>1$. For all tuples $\bar{\mathsf{b}}_1 \in \mathsf{A}^{|\bar{\mathrm{x}}_1|},\dots,\bar{\mathsf{b}}_k \in \mathsf{A}^{|\bar{\mathrm{x}}_k|}$ we set $\tau_k(\bar{\mathsf{b}}_1,\dots,\bar{\mathsf{b}}_k) := \sigma_k(\bar{\mathsf{c}}_1,\dots,\bar{\mathsf{c}}_k)$, where $\langle\bar{\mathsf{c}}_1,\dots,\bar{\mathsf{c}}_k\rangle := \alpha_{k,\langle\overline{S}_0^{(k)},\dots,\overline{S}_{k-1}^{(k)}\rangle}$ and we have

$$\begin{aligned}
S_{0,j}^{(k)} &= \mu_{0,j}\big(\bar{\mathsf{b}}_1,\dots,\bar{\mathsf{b}}_j\big) \quad \text{for every } j,\ 0<j\le k,\\
S_{1,j}^{(k)} &= \mu_{1,j}\big(\tau_1(\bar{\mathsf{b}}_1),\bar{\mathsf{b}}_2,\dots,\bar{\mathsf{b}}_j\big) \quad \text{for every } j,\ 1<j\le k,\\
&\ \ \vdots\\
S_{k-2,j}^{(k)} &= \mu_{k-2,j}\big(\tau_1(\bar{\mathsf{b}}_1),\dots,\tau_{k-2}(\bar{\mathsf{b}}_1,\dots,\bar{\mathsf{b}}_{k-2}),\bar{\mathsf{b}}_{k-1},\dots,\bar{\mathsf{b}}_j\big) \quad \text{for every } j,\ k-2<j\le k,\\
S_{k-1,k}^{(k)} &= \mu_{k-1,k}\big(\tau_1(\bar{\mathsf{b}}_1),\dots,\tau_{k-1}(\bar{\mathsf{b}}_1,\dots,\bar{\mathsf{b}}_{k-1}),\bar{\mathsf{b}}_k\big)\ ,
\end{aligned}$$

if such an $\alpha_{k,\langle\overline{S}_0^{(k)},\dots,\overline{S}_{k-1}^{(k)}\rangle}$ exists — we shall show in Claim IV that this is always the case.

Claim III: For all ℓ,k, $0\le\ell<k\le n$, we have $\mathrm{im}_\tau\big(\mu_{\ell,k}\big)\subseteq\mathrm{im}_\sigma\big(\mu_{\ell,k}\big)$.

Proof: Fix some $\mu_{\ell,k}$ and let $S\in\mathrm{im}_\tau(\mu_{\ell,k})$. Hence, there are tuples $\bar{\mathsf{b}}_1,\dots,\bar{\mathsf{b}}_k$ such that $\tau_1(\bar{\mathsf{b}}_1),\dots,\tau_k(\bar{\mathsf{b}}_1,\dots,\bar{\mathsf{b}}_k)$ are defined and we have

$$S=\mu_{\ell,k}\big(\tau_1(\bar{\mathsf{b}}_1),\dots,\tau_\ell(\bar{\mathsf{b}}_1,\dots,\bar{\mathsf{b}}_\ell),\bar{\mathsf{b}}_{\ell+1},\dots,\bar{\mathsf{b}}_k\big).$$

By definition of τ, there are representatives $\alpha_{j,\langle\overline{S}_0^{(j)},\dots,\overline{S}_{j-1}^{(j)}\rangle}=\langle\bar{\mathsf{c}}_1^{(j)},\dots,\bar{\mathsf{c}}_j^{(j)}\rangle$, $1\le j\le\ell$, for which we observe the following properties.

(a) For every i, $0\le i<\ell$, all the $\overline{S}_i^{(j)}$ are prefixes of $\overline{S}_i^{(\ell)}$. This means, if we write $\overline{S}_i^{(\ell)}$ for the sequence $S_{i,i+1},\dots,S_{i,\ell}$, we have
$S_{i,i+1}\dots S_{i,\ell} \;=\; \overline{S}_i^{(1)} S_{i,i+2}\dots S_{i,\ell} \;=\; \overline{S}_i^{(2)} S_{i,i+3}\dots S_{i,\ell} \;=\dots=\; \overline{S}_i^{(\ell-1)} S_{i,\ell} \;=\; \overline{S}_i^{(\ell)}$.

(b) For every j, $1\le j\le\ell$, we have $\tau_j(\bar{\mathsf{b}}_1,\dots,\bar{\mathsf{b}}_j)=\sigma_j(\bar{\mathsf{c}}_1^{(j)},\dots,\bar{\mathsf{c}}_j^{(j)})$.

Because of (a) and due to the construction of the $\alpha_{j,\langle\overline{S}_0^{(j)},\dots,\overline{S}_{j-1}^{(j)}\rangle}=\langle\bar{\mathsf{c}}_1^{(j)},\dots,\bar{\mathsf{c}}_j^{(j)}\rangle$, we have $\bar{\mathsf{c}}_i^{(j)}=\bar{\mathsf{c}}_i^{(j')}$ for every i, $1\le i\le\ell$, and all j,j', $1\le j,j'\le\ell$. Hence, we can write $\bar{\mathsf{c}}_1,\dots,\bar{\mathsf{c}}_\ell$ instead of $\bar{\mathsf{c}}_1^{(j)},\dots,\bar{\mathsf{c}}_1^{(j)}$ (for any j). Therefore, (b) entails

$$\begin{aligned}
S &= \mu_{\ell,k}\big(\tau_1(\bar{\mathsf{b}}_1),\dots,\tau_\ell(\bar{\mathsf{b}}_1,\dots,\bar{\mathsf{b}}_\ell),\bar{\mathsf{b}}_{\ell+1},\dots,\bar{\mathsf{b}}_k\big)\\
&= \mu_{\ell,k}\big(\sigma_1(\bar{\mathsf{c}}_1),\dots,\sigma_\ell(\bar{\mathsf{c}}_1,\dots,\bar{\mathsf{c}}_\ell),\bar{\mathsf{b}}_{\ell+1},\dots,\bar{\mathsf{b}}_k\big).
\end{aligned}$$

Consequently, $S\in\mathrm{im}_\sigma(\mu_{\ell,k})$. ◊

Claim IV: For every k, $1 \leq k \leq n$, and all tuples $\bar{\mathsf{b}}_1, \dots, \bar{\mathsf{b}}_k$ there is a representative $\alpha_{k,\langle \overline{S}_0,\dots,\overline{S}_{k-1}\rangle}$ such that

$$\begin{aligned}
S_{0,j} &= \mu_{0,j}\big(\bar{\mathsf{b}}_1, \dots, \bar{\mathsf{b}}_j\big) \quad \text{for every } j,\ 0 < j \leq k,\\
S_{1,j} &= \mu_{1,j}\big(\tau_1(\bar{\mathsf{b}}_1), \bar{\mathsf{b}}_2, \dots, \bar{\mathsf{b}}_j\big) \quad \text{for every } j,\ 1 < j \leq k,\\
&\vdots\\
S_{k-2,j} &= \mu_{k-2,j}\big(\tau_1(\bar{\mathsf{b}}_1), \dots, \tau_{k-2}(\bar{\mathsf{b}}_1, \dots, \bar{\mathsf{b}}_{k-2}), \bar{\mathsf{b}}_{k-1}, \dots, \bar{\mathsf{b}}_j\big) \quad \text{for every } j,\ k-2 < j \leq k,\\
S_{k-1,k} &= \mu_{k-1,k}\big(\tau_1(\bar{\mathsf{b}}_1), \dots, \tau_{k-1}(\bar{\mathsf{b}}_1, \dots, \bar{\mathsf{b}}_{k-1}), \bar{\mathsf{b}}_k\big)\ .
\end{aligned}$$

Proof: We proceed by induction on k.

Let $k = 1$. Consider any tuple $\bar{\mathsf{b}}_1 \in \mathsf{A}^{|\bar{\mathrm{x}}_1|}$ and set $S_0 := \mu_{0,1}\big(\bar{\mathsf{b}}_1\big)$. Hence, $S_0 \in \mathrm{im}_\sigma(\mu_{0,1})$ and we thus have defined the partition $\underline{\mathsf{A}}_{1,\langle S_0\rangle}$. Since $\bar{\mathsf{b}}_1 \in \underline{\mathsf{A}}_{1,\langle S_0\rangle}$, the set is nonempty and there is a representative $\alpha_{1,\langle S_0\rangle} \in \underline{\mathsf{A}}_{1,\langle S_0\rangle}$.

Let $k > 1$. Consider any sequence of tuples $\bar{\mathsf{b}}_1 \in \mathsf{A}^{|\bar{\mathrm{x}}_1|}, \dots, \bar{\mathsf{b}}_k \in \mathsf{A}^{|\bar{\mathrm{x}}_k|}$ and defing S_i, j as in the claim. By Claim III, we have $S_{i,j} \in \mathrm{im}_\tau(\mu_{i,j}) \subseteq \mathrm{im}_\sigma(\mu_{i,j})$ for all i, j with $0 \leq i < j \leq k$ and, therefore, we have constructed the subset $\underline{\mathsf{A}}_{k,\langle \overline{S}_0,\dots,\overline{S}_{k-1}\rangle} \subseteq \underline{\mathsf{A}}_k$ when we have been defining representatives. It remains to show that this set is not empty.

For every ℓ, $0 \leq \ell < k-1$, we set $\overline{S}^{(k-1)}_\ell := S_{\ell,\ell+1} \dots S_{\ell,k-1}$. By induction, there is a representative $\alpha_{k-1,\langle \overline{S}^{(k-1)}_0,\dots,\overline{S}^{(k-1)}_{k-2}\rangle} =: \langle \bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_{k-1}\rangle$.

As one consequence, the definition of τ entails

$$\begin{aligned}
&\tau_1(\bar{\mathsf{b}}_1) = \sigma_1(\bar{\mathsf{c}}_1) = \tau_1(\bar{\mathsf{c}}_1)\ ,\\
&\qquad\vdots\\
&\tau_{k-1}(\bar{\mathsf{b}}_1, \dots, \bar{\mathsf{b}}_{k-1}) = \sigma_{k-1}(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_{k-1}) = \tau_{k-1}(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_{k-1})\ ,
\end{aligned}$$

which entails

$$\begin{aligned}
(*)\ \ &\mu_{k-1,k}\big(\tau_1(\bar{\mathsf{c}}_1), \dots, \tau_{k-1}(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_{k-1}), \bar{\mathsf{b}}_k\big)\\
&\quad= \mu_{k-1,k}\big(\tau_1(\bar{\mathsf{b}}_1), \dots, \tau_{k-1}(\bar{\mathsf{b}}_1, \dots, \bar{\mathsf{b}}_{k-1}), \bar{\mathsf{b}}_k\big)\\
&\quad= S_{k-1,k}.
\end{aligned}$$

By definition of the $\mu_{\ell,k-1}$ and since we have $S_{0,k} \in S_{0,k-1}, \dots, S_{k-2,k} \in S_{k-2,k-1}$, the properties of $\alpha_{k-1,\langle \overline{S}^{(k-1)}_0,\dots,\overline{S}^{(k-1)}_{k-2}\rangle} = \langle \bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_{k-1}\rangle$ entail the existence of tuples $\bar{\mathsf{d}}^{(0)}_k, \dots, \bar{\mathsf{d}}^{(k-2)}_k \in \mathsf{A}^{|\bar{\mathrm{x}}_k|}$ such that

$$\begin{aligned}
&\mu_{0,k}\big(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}^{(0)}_k\big) = S_{0,k}\ ,\\
&\mu_{1,k}\big(\tau_1(\bar{\mathsf{c}}_1), \bar{\mathsf{c}}_2, \dots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}^{(1)}_k\big) = S_{1,k}\ ,\\
&\qquad\vdots\\
&\mu_{k-2,k}\big(\tau_1(\bar{\mathsf{c}}_1), \dots, \tau_{k-2}(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_{k-2}), \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}^{(k-2)}_k\big) = S_{k-2,k}\ ,\ \text{and}\\
&\mu_{k-1,k}\big(\tau_1(\bar{\mathsf{c}}_1), \dots, \tau_{k-2}(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_{k-2}), \tau_{k-1}(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_{k-1}), \bar{\mathsf{b}}_k\big) = S_{k-1,k}
\end{aligned}$$

(the last equation follows from $(*)$).

Due to $S_{0,k} \in \mathcal{P}^{n-k+1}\mathrm{At}_0, \dots, S_{k-2,k} \in \mathcal{P}^{n-k+1}\mathrm{At}_{k-2}$, and $S_{k-1,k} \in \mathcal{P}^{n-k+1}\mathrm{At}_{k-1}$, Condition (ii) of Definition 3.4.1 entails pairwise disjointness of the sets $\mathrm{vars}(S_{0,k}) \cap \bar{\mathrm{x}}, \dots, \mathrm{vars}(S_{k-2,k}) \cap \bar{\mathrm{x}}$, and $\mathrm{vars}(S_{k-1,k}) \cap \bar{\mathrm{x}}$. Consequently, we can define a new tuple $\bar{\mathsf{d}}'_k$ by setting

$$\mathsf{d}'_{k,i} := \begin{cases} \mathsf{d}^{(j)}_{k,i} & \text{if } x_{k,i} \in \mathrm{vars}(S_{j,k}) \cap \bar{\mathrm{x}} \text{ with } j < k-1,\\ \mathsf{b}_{k,i} & \text{if } x_{k,i} \in \mathrm{vars}(S_{k-1,k}) \cap \bar{\mathrm{x}},\\ \mathsf{b}_{k,i} & \text{otherwise (we could use any value here).} \end{cases}$$

Due to the pairwise disjointness of the sets $\mathrm{vars}(S_{0,k}) \cap \bar{\mathrm{x}}, \ldots, \mathrm{vars}(S_{k-1,k}) \cap \bar{\mathrm{x}}$, Claim II implies that for every ℓ, $0 \leq \ell < k-1$,

$$\begin{aligned}&\mu_{\ell,k}\big(\tau_1(\bar{\mathsf{c}}_1), \ldots, \tau_\ell(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_\ell), \bar{\mathsf{c}}_{\ell+1}, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}'_k\big)\\ &\quad= \mu_{\ell,k}\big(\tau_1(\bar{\mathsf{c}}_1), \ldots, \tau_\ell(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_\ell), \bar{\mathsf{c}}_{\ell+1}, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}^{(\ell)}_k\big)\\ &\quad= S_{\ell,k}\end{aligned}$$

and

$$\begin{aligned}&\mu_{k-1,k}\big(\tau_1(\bar{\mathsf{c}}_1), \ldots, \tau_{k-1}(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}), \bar{\mathsf{d}}'_k\big)\\ &\quad= \mu_{k-1,k}\big(\tau_1(\bar{\mathsf{c}}_1), \ldots, \tau_{k-1}(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}), \bar{\mathsf{b}}_k\big)\\ &\quad= S_{k-1,k}.\end{aligned}$$

Consequently, the set $\underline{\mathsf{A}}_{k,\langle \overline{S}_0,\ldots,\overline{S}_{k-1}\rangle}$ contains at least the tuple $\langle \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}'_k\rangle$. Therefore, there exists some representative $\alpha_{k,\langle \overline{S}_0,\ldots,\overline{S}_{k-1}\rangle} \in \underline{\mathsf{A}}_{k,\langle \overline{S}_0,\ldots,\overline{S}_{k-1}\rangle}$. ◊

Claim V: τ is μ-uniform.

Proof: By construction of τ. ◊

Now let $S \in \mathrm{Out}_\tau$. Then, there exist tuples $\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_n$ such that $S = \mathrm{out}_\tau(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_n)$. We partition S into sets $S_0 := S \cap \mathrm{At}_0, \ldots, S_n := S \cap \mathrm{At}_n$ and thus obtain the fingerprints $S_\ell = \mu_{\ell,n}\big(\tau_1(\bar{\mathsf{b}}_1), \ldots, \tau_\ell(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_\ell), \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_n\big) \subseteq \mathrm{At}_\ell$ for every ℓ, $0 \leq \ell < n$. Claim IV guarantees the existence of some representative $\alpha_{n,\langle \overline{S}'_0,\ldots,\overline{S}'_{n-1}\rangle} = \langle \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_n\rangle$ such that $S_\ell = \mu_{\ell,n}\big(\sigma_1(\bar{\mathsf{c}}_1), \ldots, \sigma_\ell(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_\ell), \bar{\mathsf{c}}_{\ell+1}, \ldots, \bar{\mathsf{c}}_n\big)$ for every ℓ, $0 \leq \ell < n$.

Consider any $A(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_\ell, \bar{\mathrm{x}}_{\ell+1}, \ldots, \bar{\mathrm{x}}_n) \in \mathrm{At}$, and fix the ℓ for which $A \in \mathrm{At}_\ell$. We distinguish two cases. Suppose that $\ell < n$. The definition of $\alpha_{n,\langle \overline{S}'_0,\ldots,\overline{S}'_{n-1}\rangle}$ and the fingerprint functions $\mu_{\ell,n}$ entail that $A \in S_\ell$ if and only if

$$\mathcal{A} \models A\big(\tau_1(\bar{\mathsf{b}}_1), \ldots, \tau_\ell(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_\ell), \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_n\big)$$

if and only if

$$\mathcal{A} \models A\big(\sigma_1(\bar{\mathsf{c}}_1), \ldots, \sigma_\ell(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_\ell), \bar{\mathsf{c}}_{\ell+1}, \ldots, \bar{\mathsf{c}}_n\big).$$

In case of $\ell = n$, we have $A(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n) \in S_n$ if and only if

$$\mathcal{A} \models A\big(\tau_1(\bar{\mathsf{b}}_1), \ldots, \tau_n(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_n)\big)$$

if and only if

$$\mathcal{A} \models A\big(\sigma_1(\bar{\mathsf{c}}_1), \ldots, \sigma_n(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_n)\big).$$

In both cases, we get $A \in \mathrm{out}_\tau(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_n)$ if and only if $A \in \mathrm{out}_\sigma(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_n)$. Consequently, we have $S = \mathrm{out}_\tau(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_n) = \mathrm{out}_\sigma(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_n) \in \mathrm{Out}_\sigma$. Altogether, it follows that $\mathrm{Out}_\tau \subseteq \mathrm{Out}_\sigma$. □

Corollary 4.2.8. *If there is a satisfying strategy σ for φ, then there is also a μ-uniform strategy τ that is satisfying for φ (under $\mathcal{A}$).*

Proof. Let σ be a satisfying strategy for φ. By Lemma 4.2.7, there is a μ-uniform strategy τ such that for every $S \in \mathrm{Out}_\tau$ we have $S \in \mathrm{Out}_\sigma$. Since σ is satisfying for φ, every $S \in \mathrm{Out}_\sigma$ can be conceived as an assignment of truth values to the atoms in φ's quantifier-free part ψ such that ψ is satisfied. If this applies to every $S \in \mathrm{Out}_\sigma$, then it certainly applies to any $S \in \mathrm{Out}_\tau \subseteq \mathrm{Out}_\sigma$. Therefore, τ is also satisfying for φ. □

The guaranteed existence of μ-uniform satisfying strategies for all models of GBSR sentences confirms that all dependences in GBSR sentences are weak. On the other hand, it entails that GBSR enjoys the finite model property. In order to formulate the induced bound regarding the cardinality of small models accurately, we introduce another notion of *degree* for GBSR sentences that is suitable for this purpose. This time the degree is based on co-occurrences of universally quantified variables in atoms. This notion complements the notion introduced in Section 3.5 (Definition 3.5.1).

$\partial_\forall(\varphi)$

Definition 4.2.9 (Degree of interaction of universal variables)**.** *We denote by $\partial_\forall(\varphi)$ the* degree of interaction of universal variables *in φ, defined to be the smallest nonnegative integer meeting the following condition. For every At_i, $0 \leq i < n$, there are at most $\partial_\forall(\varphi)$ pairwise distinct indices $i+1 < j_1 < \ldots < j_{\partial_\forall(\varphi)} \leq n$ such that $\bar{\mathrm{x}}_{j_\ell} \cap \mathrm{vars}(\mathrm{At}_i) \neq \emptyset$.*

Notice that we have $0 \leq \partial_\forall(\varphi) < n$. Moreover, in this definition the degree $\partial_\forall(\varphi)$ implicitly depends on the currently chosen partition of At into the sets $\mathrm{At}_0, \ldots, \mathrm{At}_n$. Consider, for instance, an MFO sentence φ_{MFO}. We could partition its atoms into two parts $\mathrm{At}_0, \mathrm{At}_n$, where At_0 contains all atoms with a universally quantified variable and At_n comprises all other atoms. Clearly, At_0 will cause the highest possible degree for φ_{MFO}, since all universal variables occur in At_0. We get a lower degree, if we partition At as follows. For every i, $0 \leq i < n$, we set $\mathrm{At}_i := \{P(x) \in \mathrm{At} \mid x \in \bar{\mathrm{x}}_{i+1}\}$, and the set At_n again contains the rest of the atoms. This partition induces the potentially much lower degree $\partial_\forall(\varphi_{\mathrm{MFO}}) = 0$. Although this dependence on the current partition of At could be eliminated by minimizing the degree over all possible partitions, compare also Definition 3.5.1, the weaker notion given in Definition 4.2.9 suffices for the moment.

Lemma 4.2.10. *If there is a satisfying μ-uniform strategy σ for φ, then the substructure $\mathcal{A}|_\sigma$ of $\mathcal{A}$ is a model of φ. Moreover, $\mathcal{A}|_\sigma$ comprises at most $n \cdot |\bar{\mathrm{y}}| \cdot \big(2^{\uparrow \partial_\forall(\varphi)+1}(|\mathrm{At}|)\big)^{n^2}$ elements.*

Proof. We start with two preliminary results.

Claim I: Let ℓ, k be two integers with $0 \leq \ell < k < n$. For all tuples $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k$ with $\bar{\mathsf{a}}_i \in \mathsf{A}^{|\bar{\mathrm{y}}_i|}$ and $\bar{\mathsf{b}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}_i|}$ for every i we observe that, if $\mathrm{vars}(\mathrm{At}_\ell) \cap \bar{\mathrm{x}}_{k+1} = \emptyset$, then $\big|\mu_{\ell,k}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k)\big| = 1$ and, consequently, $|\mathrm{im}_\sigma(\mu_{\ell,k})| \leq |\mathrm{im}_\sigma(\mu_{\ell,k+1})|$.

Proof: Suppose there are sets $S_1, S_2 \in \mu_{\ell,k}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k)$ that are distinct. Hence, there are tuples $\bar{\mathsf{c}}_{k+1}, \bar{\mathsf{d}}_{k+1} \in \mathsf{A}^{|\bar{\mathrm{x}}_{k+1}|}$ such that $S_1 = \mu_{\ell,k+1}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{c}}_{k+1})$ and $S_2 = \mu_{\ell,k+1}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{d}}_{k+1})$. But since $\bar{\mathrm{x}}_{k+1} \cap \mathrm{vars}(\mathrm{At}_\ell) = \emptyset$, Claim II from the proof of Lemma 4.2.7 entails $S_1 = S_2$. This contradicts our assumption that S_1 and S_2 are distinct. Consequently, $\mu_{\ell,k}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k)$ contains at most one set.

It remains to show that $\mu_{\ell,k}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k)$ is nonempty. This is easily done by induction on $k < n$, starting from $k = n - 1$. ◊

Claim II: Let ℓ, k be two integers with $0 \leq \ell < k < n$. We have $|\mathrm{im}_\sigma(\mu_{\ell,k})| \leq 2^{|\mathrm{im}_\sigma(\mu_{\ell,k+1})|}$.

Proof: For all tuples $\bar{\mathsf{b}}_1, \ldots, \ldots, \bar{\mathsf{b}}_k$ with $\bar{\mathsf{b}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}_i|}$ and for every

$$S \in \mu_{\ell,k}(\sigma_1(\bar{\mathsf{b}}_1), \ldots, \sigma_\ell(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_\ell), \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k)$$

we know that $S = \mu_{\ell,k+1}(\sigma_1(\bar{\mathsf{b}}_1), \ldots, \sigma_\ell(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_\ell), \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{c}}_{k+1})$ for some tuple $\bar{\mathsf{c}}_{k+1}$. Hence, $\mu_{\ell,k}(\sigma_1(\bar{\mathsf{b}}_1), \ldots, \sigma_\ell(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_\ell), \bar{\mathsf{b}}_{\ell+1}, \ldots, \bar{\mathsf{b}}_k) \subseteq \mathrm{im}_\sigma(\mu_{\ell,k+1})$. ◊

Due to Claim I and Claim II, we observe that

(∗) for all integers ℓ, k with $0 \leq \ell < k \leq n$ we obtain $|\mathrm{im}_\sigma(\mu_{\ell,k})| \leq 2^{\uparrow \partial_\forall(\varphi)+1}(|\mathrm{At}_\ell|)$.

Let $\mathcal{T}_\sigma$ be the *target set* of σ, defined by $\mathcal{T}_\sigma := \bigcup_{k=1}^n \mathcal{T}_k$, where $\mathcal{T}_\sigma$, $\mathcal{T}_k$

$$\mathcal{T}_k := \{\mathsf{a} \in \mathsf{A} \mid \text{ there are tuples } \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_k \text{ such that } \sigma_k(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_k) = \langle \ldots, \mathsf{a}, \ldots \rangle\} .$$

Notice that $\mathcal{T}_\sigma$ coincides with the domain of $\mathcal{A}|_\sigma$. Since σ is μ-uniform, we know that $\mathcal{T}_\sigma$ is a finite set. By definition of the fingerprint functions $\mu_{\ell,k}$, we get the following upper bounds, where we write $\underline{\mathrm{im}}_\sigma(\mu_{i,j})$ to abbreviate $\mathrm{im}_\sigma(\mu_{i,i+1}) \times \mathrm{im}_\sigma(\mu_{i,i+2}) \times \ldots \times \mathrm{im}_\sigma(\mu_{i,j})$ for all i, j, $0 \leq i < j \leq n$. $\underline{\mathrm{im}}_\sigma(\mu_{i,j})$

$$\begin{aligned}
\big|\mathcal{T}_1\big| &\leq |\bar{\mathrm{y}}_1| \cdot \big|\underline{\mathrm{im}}_\sigma(\mu_{0,1})\big| \leq |\bar{\mathrm{y}}_1| \cdot 2^{\uparrow n}(|\mathrm{At}_0|) \leq |\bar{\mathrm{y}}_1| \cdot 2^{\uparrow n}(|\mathrm{At}|),\\
\big|\mathcal{T}_2\big| &\leq |\bar{\mathrm{y}}_2| \cdot \big|\underline{\mathrm{im}}_\sigma(\mu_{0,2}) \times \underline{\mathrm{im}}_\sigma(\mu_{1,2})\big| \leq |\bar{\mathrm{y}}_2| \cdot 2^{\uparrow n}(|\mathrm{At}_0|) \cdot 2^{\uparrow n-1}(|\mathrm{At}_0|) \cdot 2^{\uparrow n-1}(|\mathrm{At}_1|)\\
&\leq |\bar{\mathrm{y}}_2| \cdot \big(2^{\uparrow n}(|\mathrm{At}|)\big)^3,\\
&\vdots\\
\big|\mathcal{T}_n\big| &\leq |\bar{\mathrm{y}}_n| \cdot \big|\underline{\mathrm{im}}_\sigma(\mu_{0,n}) \times \ldots \times \underline{\mathrm{im}}_\sigma(\mu_{n-1,n})\big| \leq |\bar{\mathrm{y}}_n| \cdot \textstyle\prod_{i=0}^{n-1} \prod_{j=i}^{n-1} 2^{\uparrow n-j}(|\mathrm{At}_j|)\\
&\leq |\bar{\mathrm{y}}_n| \cdot \big(2^{\uparrow n}(|\mathrm{At}|)\big)^{n^2}.
\end{aligned}$$

When we combine these bounds with the bound formulated in (∗), it follows that $\mathcal{T}_\sigma$ contains

at most $\sum_{\ell=1}^{n} |\bar{y}_\ell| \cdot \prod_{i=0}^{n-1} \prod_{j=i}^{n-1} 2^{\uparrow \min(\partial_\forall(\varphi)+1, n-j)}(|\mathrm{At}_j|) \;\leq\; n \cdot |\bar{y}| \cdot \left(2^{\uparrow \partial_\forall(\varphi)+1}(|\mathrm{At}|)\right)^{n^2}$ domain elements.

Now consider the structure $\mathcal{A}|_\sigma$. We have already noted that $\mathsf{A}|_\sigma = \mathcal{T}_\sigma$. Hence, the above bound also applies to the number of elements in $\mathcal{A}|_\sigma$'s domain. By Lemma 4.2.3, $\mathcal{A}|_\sigma$ is a finite model of φ. □

Theorem 4.2.11. *Every satisfiable GBSR sentence $\varphi := \forall \bar{x}_1 \exists \bar{y}_1 \ldots \forall \bar{x}_n \exists \bar{y}_n . \psi$ with quantifier-free ψ has a model with at most $\mathrm{len}(\varphi)^2 \cdot \left(2^{\uparrow \partial_\forall(\varphi)+1}(\mathrm{len}(\varphi))\right)^{n^2}$ domain elements. More precisely, every model $\mathcal{A} \models \varphi$ contains a substructure that satisfies φ and whose domain size is bounded as stated.*

It is worth noticing that Theorem 4.2.11 states a small model property for GBSR that is, in contrast to Corollary 3.5.4, not based on the equivalence with BSR but is solely inferred via the analysis of weak dependences and the existence of μ-uniform satisfying strategies for satisfiable sentences. Moreover, when applied to SF sentences, the bounds in Theorem 4.2.11 might be smaller than the bounds given in Theorem 3.2.6, or vice versa, as for any SF sentence ψ the two degrees $\partial_\exists(\psi)$ and $\partial_\forall(\psi)$ are largely independent from each other.

Remark 4.2.12. *The second part of Theorem 4.2.11 emphasizes the fact that all models of a GBSR sentence φ contain a finite substructure that satisfies φ as well. This observation is trivial for BSR sentences, and, via the equivalence of GBSR and BSR, the property follows already from syntactic arguments.*

What is also easy to verify for any BSR Σ-sentence ψ is the following. Let $\mathcal{A}_1, \ldots, \mathcal{A}_n$ be any chain of Σ-structures where every $\mathcal{A}_{i+1}$ is a proper substructure of $\mathcal{A}_i$. If $\mathcal{A}_1$ and $\mathcal{A}_n$ are models of ψ, then each and every $\mathcal{A}_i$ in the chain satisfies ψ. In general, there is no guarantee that non-trivial chains of such satisfying (sub)structures can be extended to the right until $\mathcal{A}_n$ comprises only a single domain element. The Łoś–Tarski Theorem [Tar54, Łoś55] (see also Theorem 6.5.4 in [Hod93]) stipulates that this is possible if and only if ψ is equivalent to some $\forall^*$-sentence.

In [SC10], Section 3, the above property of BSR sentences is described as "preservation under substructures modulo a bounded 'core'*" and the close relation to the Łoś–Tarski is pointed out. Moreover, the authors propose a generalization of this property as a semantic characterization of BSR.*

By Theorems 3.5.3 and 4.2.1, all of the above said also applies to GBSR and the class of first-order sentences in which all dependences of existentially quantified variables on universally quantified variables are weak.

4.3 GAF Sentences and the Existence of Semi-Uniform Winning Strategies

In the present section we investigate the dependences occurring in GAF sentences. Evidently, strong dependences may occur, like the simple Ackermann sentence $\forall x \exists y . E(x, y)$ illustrates. Its Skolemized variant $\forall x . E(x, f(x))$ has a model $\mathcal{A}$ with $\mathsf{A} := \{0, 1, 2, \ldots\}$, $E^\mathcal{A} := \{\langle k, k+1 \rangle \mid k \geq 0\}$, and $f^\mathcal{A}(k) = k+1$ for every k, where altering $f^\mathcal{A}$ to any function with a finite image does not result in a model of the sentence. Nevertheless, we will see that strong dependences in GAF sentences are finitely controllable, i.e. infinite models cannot be enforced. This coincides with our earlier observation that GAF enjoys the finite model property. In contrast to sentences with only weak dependences, not every model of a given GAF sentence admits satisfying strategies with finite images.

Consider any GAF sentence $\varphi := \forall \bar{x}_1 \exists \bar{y}_1 \bar{u}_1 \ldots \forall \bar{x}_n \exists \bar{y}_n \bar{u}_n . \psi$. As usual, we set $\bar{x} := \bar{x}_1 \cup \ldots \cup \bar{x}_n$, $\bar{y} := \bar{y}_1 \cup \ldots \cup \bar{y}_n$, and $\bar{u} := \bar{u}_1 \cup \ldots \cup \bar{u}_n$. Recall that, according to Definition 3.7.1, every variable $u \in \bar{u}$ that occurs in φ is associated with exactly one reference variable $x \in \bar{x}$, determined by the set At_x in which u occurs. When we investigate the occurring dependences in the framework of model-checking games, we observe that the only strong dependences in φ occur between variables u and their respective reference variable x. All other dependences in φ are weak, in particular the

ones between any $y \in \bar{\mathrm{y}}$ and any variables from $\bar{\mathrm{x}}$. Satisfying strategies may need infinitely many options for variables from $\bar{\mathrm{u}}$ to appropriately respond to all possible values of variables from $\bar{\mathrm{x}}$, just like in the case of the above described model $\mathcal{A}$ for $\forall x \exists y.\, E(x,y)$. Example 4.3.1 illustrates this observation in a slightly more involved setting. This means, we cannot always find *uniform* satisfying strategies in the sense of Definition 4.2.6 with respect to appropriate *fingerprints* for GAF. In order to compensate for this, we introduce the weaker notion of *semi-uniform strategies*, which exist for every model of φ. Although semi-uniform satisfying strategies do not directly induce finite models, they do facilitate the construction of finite models.

We base our considerations on a fixed GAF sentence $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \bar{\mathrm{u}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n \bar{\mathrm{u}}_n.\, \psi$ in standard form with quantifier-free ψ. The set At and the tuples $\bar{\mathrm{x}}, \bar{\mathrm{y}}, \bar{\mathrm{u}}$ are defined as in Definition 3.7.1. Then, At can be partitioned into sets At_0 and At_x, $x \in \bar{\mathrm{x}}$, in accordance with Definition 3.7.1. In addition, we define the set $U_x := \mathrm{vars}(\mathrm{At}_x) \cap \bar{\mathrm{u}}$ for every $x \in \bar{\mathrm{x}}$. Recall that we have $\mathrm{vars}(\mathrm{At}_x) \subseteq \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_{\mathrm{idx}(x)-1} \cup \{x\} \cup \bar{\mathrm{u}}_{\mathrm{idx}(x)} \cup \ldots \cup \bar{\mathrm{u}}_n$ for every x, and that for any two distinct $x, x' \in \bar{\mathrm{x}}$ we have $U_x \cap U_{x'} = \emptyset$.

φ

U_x

Let $\mathcal{A}$ be any structure over the vocabulary of φ. We adapt the definition of *strategy* and related notions from Section 4.2 as follows. In the GAF setting, a strategy σ is a tuple of mappings $\langle \sigma_1, \ldots, \sigma_n \rangle$ with the signatures $\sigma_i : \mathsf{A}^{|\bar{\mathrm{x}}_1|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{x}}_i|} \to \mathsf{A}^{|\bar{\mathrm{y}}_i|} \times \mathsf{A}^{|\bar{\mathrm{u}}_i|}$. For convenience, we sometimes split σ_i into two parts: a $\bar{\mathrm{y}}_i$-part $\sigma_i^1 : \mathsf{A}^{|\bar{\mathrm{x}}_1|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{x}}_i|} \to \mathsf{A}^{|\bar{\mathrm{y}}_i|}$ and a $\bar{\mathrm{u}}_i$-part $\sigma_i^2 : \mathsf{A}^{|\bar{\mathrm{x}}_1|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{x}}_i|} \to \mathsf{A}^{|\bar{\mathrm{u}}_i|}$. A strategy σ is *satisfying for* φ if

$\mathcal{A}$

$$\mathcal{A}, [\bar{\mathrm{x}}_1 \mapsto \bar{\mathsf{a}}_1, \ldots, \bar{\mathrm{x}}_n \mapsto \bar{\mathsf{a}}_n, \bar{\mathrm{y}}_1 \bar{\mathrm{u}}_1 \mapsto \sigma_1(\bar{\mathsf{a}}_1), \ldots, \bar{\mathrm{y}}_n \bar{\mathrm{u}}_n \mapsto \sigma_n(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n)] \models \psi$$

holds for every choice of tuples $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n$ of appropriate length. The other related notions, such as *outcomes*, are adapted accordingly.

Example 4.3.1. *Consider the GAF sentence*

$$\varphi := \exists z \forall x \exists y_1 y_2.\, Q(z) \wedge \neg R(x,x) \wedge R(x,y_1) \wedge Q(y_1) \wedge R(x,y_2) \wedge \neg Q(y_2)\ .$$

We partition the set of atoms of φ into $\mathrm{At}_0 := \{Q(z)\}$ and $\mathrm{At}_x := \{R(x,x), R(x,y_1), Q(y_1), R(x,y_2), Q(y_2)\}$. One possible model $\mathcal{A}$ is given by $\mathsf{A} := \{0,1,2,3,\ldots\}$, $Q^{\mathcal{A}} := \{0,1,3,5,\ldots\}$, $R^{\mathcal{A}} := \big\{\langle i, i+1\rangle, \langle i, i+2\rangle \;\big|\; i \text{ even}\big\} \cup \big\{\langle i, i+2\rangle, \langle i, i+3\rangle \;\big|\; i \text{ odd}\big\}$. A satisfying strategy is the canonical σ with $\sigma_1^1() := 0$, where $()$ denotes the empty list of arguments, and $\sigma_2^2(i) := \langle i+1, i+2\rangle$ for even i, $\sigma_2^2(i) := \langle i+2, i+3\rangle$ for odd i. The model $\mathcal{A}$ is depicted in Figure 4.1, and the strategy σ is indicated by the arrows together with their annotation. Notice that none of the finite substructures of $\mathcal{A}$ is a model of φ. Hence, there is no satisfying strategy with a finite image.

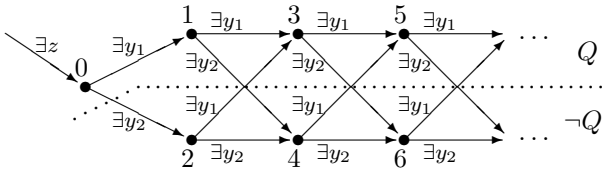


Figure 4.1: Illustration of the structure $\mathcal{A}$. An arrow from a to b indicates $\langle \mathsf{a}, \mathsf{b}\rangle \in R^{\mathcal{A}}$. The annotated existential quantifiers indicate which elements could be selected by a satisfying strategy. All domain elements above the dotted line belong to $Q^{\mathcal{A}}$, the elements below do not.

A fingerprint characterizes a class of tuples of domain elements that cannot be distinguished by a given GAF sentence. Again, there will be only finitely many such fingerprints.

Definition 4.3.2 (Fingerprint functions $\lambda_{x,\ell}$)**.** *We define the family of* fingerprint functions $\lambda_{x,\ell}$ *as follows. For every k, $1 \le k \le n$, and every $x \in \bar{\mathrm{x}}_k$ we define the mappings*

$\lambda_{x,\ell}$

$\lambda_{x,n} : \mathsf{A}^{|\bar{\mathrm{y}}_1|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{y}}_{k-1}|} \times \mathsf{A} \times \mathsf{A}^{|\bar{\mathrm{u}}_k|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{u}}_n|} \to \mathcal{P}\mathrm{At}_x$ *such that for every atom $A(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_{k-1}, x, \bar{\mathrm{u}}_k, \ldots, \bar{\mathrm{u}}_n) \in \mathrm{At}_x$ we have $A \in \lambda_{x,n}(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \mathsf{a}, \bar{\mathsf{c}}_k, \ldots, \bar{\mathsf{c}}_n)$ if and only if $\mathcal{A} \models A(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \mathsf{a}, \bar{\mathsf{c}}_k, \ldots, \bar{\mathsf{c}}_n)$;*

$\lambda_{x,n-1} : \mathsf{A}^{|\bar{\mathsf{y}}_1|} \times \ldots \times \mathsf{A}^{|\bar{\mathsf{y}}_{k-1}|} \times \mathsf{A} \times \mathsf{A}^{|\bar{\mathsf{u}}_k|} \times \ldots \times \mathsf{A}^{|\bar{\mathsf{u}}_{n-1}|} \to \mathcal{P}^2\mathrm{At}_x$ *such that for every* $S \in \mathcal{P}\mathrm{At}_x$ *we have* $S \in \lambda_{x,n-1}(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \mathsf{a}, \bar{\mathsf{c}}_k, \ldots, \bar{\mathsf{c}}_{n-1})$ *if and only if there exists some* $\bar{\mathsf{c}}_n$ *for which* $\lambda_{x,n}(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \mathsf{a}, \bar{\mathsf{c}}_k, \ldots, \bar{\mathsf{c}}_{n-1}, \bar{\mathsf{c}}_n) = S$*;*

$\vdots$

$\lambda_{x,k} : \mathsf{A}^{|\bar{\mathsf{y}}_1|} \times \ldots \times \mathsf{A}^{|\bar{\mathsf{y}}_{k-1}|} \times \mathsf{A} \times \mathsf{A}^{|\bar{\mathsf{u}}_k|} \to \mathcal{P}^{n-k+1}\mathrm{At}_x$ *such that for every* $S \in \mathcal{P}^{n-k}\mathrm{At}_x$ *we have* $S \in \lambda_{x,k}(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \mathsf{a}, \bar{\mathsf{c}}_k)$ *if and only if there is some* $\bar{\mathsf{c}}_{k+1}$ *for which* $\lambda_{x,k+1}(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \mathsf{a}, \bar{\mathsf{c}}_k, \bar{\mathsf{c}}_{k+1}) = S$*;*

$\lambda_{x,0} : \mathsf{A}^{|\bar{\mathsf{y}}_1|} \times \ldots \times \mathsf{A}^{|\bar{\mathsf{y}}_{k-1}|} \times \mathsf{A} \to \mathcal{P}^{n-k+2}\mathrm{At}_x$ *such that for every* $S \in \mathcal{P}^{n-k+1}\mathrm{At}_x$ *we have* $S \in \lambda_{x,0}(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \mathsf{a})$ *if and only if there exists some* $\bar{\mathsf{c}}_k$ *for which* $\lambda_{x,k}(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \mathsf{a}, \bar{\mathsf{c}}_k) = S$.

$\mathrm{im}_\sigma(\lambda_{x,0})$ *For every* $x \in \bar{\mathrm{x}}_k$ *we define the* image *of* $\lambda_{x,0}$ *under strategy* σ *by*

$$\mathrm{im}_\sigma(\lambda_{x,0}) := \big\{\lambda_{x,0}\big(\sigma^1_1(\bar{\mathsf{a}}_1), \ldots, \sigma^1_{k-1}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{k-1}), \mathsf{a}\big) \;\big|\; \bar{\mathsf{a}}_1 \in \mathsf{A}^{|\bar{\mathrm{x}}_1|}, \ldots, \bar{\mathsf{a}}_{k-1} \in \mathsf{A}^{|\bar{\mathrm{x}}_{k-1}|}, \mathsf{a} \in \mathsf{A}\big\}.$$

$\underline{\lambda}_{k,\ell}$ *The notation* $\underline{\lambda}_{k,\ell}(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \bar{\mathsf{a}}, \bar{\mathsf{c}}_k, \ldots, \bar{\mathsf{c}}_\ell)$ *abbreviates the tuple*

$$\big\langle \lambda_{x_1,\ell}(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \mathsf{a}_1, \bar{\mathsf{c}}_k, \ldots, \bar{\mathsf{c}}_\ell), \ldots, \lambda_{x_m,\ell}(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \mathsf{a}_m, \bar{\mathsf{c}}_k, \ldots, \bar{\mathsf{c}}_\ell)\big\rangle,$$

where $\langle x_1, \ldots, x_m\rangle := \bar{\mathrm{x}}_k$ *and* $\langle \mathsf{a}_1, \ldots, \mathsf{a}_m\rangle := \bar{\mathsf{a}}$*. For* $k > 0$ *we denote the* image *of* $\underline{\lambda}_{k,0}$ *under* σ *by*

$$\mathrm{im}_\sigma(\underline{\lambda}_{k,0}) := \big\{\underline{\lambda}_{k,0}\big(\sigma^1_1(\bar{\mathsf{a}}_1), \ldots, \sigma^1_{k-1}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{k-1}), \bar{\mathsf{a}}_k\big) \;\big|\; \bar{\mathsf{a}}_1 \in \mathsf{A}^{|\bar{\mathrm{x}}_1|}, \ldots, \bar{\mathsf{a}}_k \in \mathsf{A}^{|\bar{\mathrm{x}}_k|}\big\}.$$

Definition 4.3.3. *(*λ*-semi-uniformity) A strategy* $\sigma = \langle \sigma_1, \ldots, \sigma_n\rangle$ *is* λ-semi-uniform *if for every* ℓ*,* $1 \leq \ell \leq n$*, the following property holds. For all tuples* $\bar{\mathsf{a}}_1, \bar{\mathsf{a}}'_1 \in \mathsf{A}^{|\bar{\mathrm{x}}_1|}, \ldots, \bar{\mathsf{a}}_\ell, \bar{\mathsf{a}}'_\ell \in \mathsf{A}^{|\bar{\mathrm{x}}_\ell|}$ *we have*

(a) $\sigma^1_\ell(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell) = \sigma^1_\ell(\bar{\mathsf{a}}'_1, \ldots, \bar{\mathsf{a}}'_\ell)$ *and*

(b) for every k *with* $1 \leq k \leq \ell$ *we have*

$$\begin{aligned}&\underline{\lambda}_{k,\ell}\big(\sigma^1_1(\bar{\mathsf{a}}_1), \ldots, \sigma^1_{k-1}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{k-1}), \bar{\mathsf{a}}_k, \sigma^2_k(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k), \ldots, \sigma^2_\ell(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell)\big)\\ &= \underline{\lambda}_{k,\ell}\big(\sigma^1_1(\bar{\mathsf{a}}'_1), \ldots, \sigma^1_{k-1}(\bar{\mathsf{a}}'_1, \ldots, \bar{\mathsf{a}}'_{k-1}), \bar{\mathsf{a}}'_k, \sigma^2_k(\bar{\mathsf{a}}'_1, \ldots, \bar{\mathsf{a}}'_k), \ldots, \sigma^2_\ell(\bar{\mathsf{a}}'_1, \ldots, \bar{\mathsf{a}}'_\ell)\big)\ ,\end{aligned}$$

whenever all of the following conditions are satisfied:

$$\begin{aligned}\underline{\lambda}_{1,0}\big(\bar{\mathsf{a}}_1\big) &= \underline{\lambda}_{1,0}\big(\bar{\mathsf{a}}'_1\big)\ ,\\ \underline{\lambda}_{2,0}\big(\sigma^1_1(\bar{\mathsf{a}}_1), \bar{\mathsf{a}}_2\big) &= \underline{\lambda}_{2,0}\big(\sigma^1_1(\bar{\mathsf{a}}'_1), \bar{\mathsf{a}}'_2\big)\ ,\\ &\vdots\\ \underline{\lambda}_{\ell,0}\big(\sigma^1_1(\bar{\mathsf{a}}_1), \ldots, \sigma^1_{\ell-1}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{\ell-1}), \bar{\mathsf{a}}_\ell\big) &= \underline{\lambda}_{\ell,0}\big(\sigma^1_1(\bar{\mathsf{a}}'_1), \ldots, \sigma^1_{\ell-1}(\bar{\mathsf{a}}'_1, \ldots, \bar{\mathsf{a}}'_{\ell-1}), \bar{\mathsf{a}}'_\ell\big)\ .\end{aligned}$$

Consider a λ-semi-uniform strategy σ. The images of the mappings σ^1_k are finite. This indicates that every $y \in \bar{\mathrm{y}}$ is only subject to weak dependences. For the mappings σ^2_k the situation is different. This can be observed in Example 4.3.1, for instance, where σ is indeed λ-semi-uniform. Hence, λ-semi-uniform strategies do not necessarily induce finite substructures — in contrast to μ-uniform strategies. We shall see later, however, that a model $\mathcal{A}$ of φ and a λ-semi-uniform strategy σ can be used as blueprint for constructing a finite model $\mathcal{B}$ for φ equipped with a satisfying λ-semi-uniform strategy τ that has a finite image.

Lemma 4.3.4. *For every strategy* $\sigma = \langle \sigma_1, \ldots, \sigma_n\rangle$ *there is a* λ*-semi-uniform strategy* $\tau = \langle \tau_1, \ldots, \tau_n\rangle$ *with* $\mathrm{Out}_\tau \subseteq \mathrm{Out}_\sigma$.

Proof. We start with two preliminary results.

Claim I: Let k be any positive integer with $k \leq n$ and let $\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \bar{\mathsf{a}}_k$ be tuples with $\bar{\mathsf{b}}_i \in \mathsf{A}^{|\bar{\mathrm{y}}_i|}$ for every i and $\bar{\mathsf{a}}_k \in \mathsf{A}^{|\bar{\mathrm{x}}_k|}$. Let $\langle S_1, \ldots, S_{|\bar{\mathrm{x}}_k|}\rangle := \underline{\lambda}_{k,0}(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \bar{\mathsf{a}}_k)$. Consider any fingerprint $T = \langle S'_1, \ldots, S'_{|\bar{\mathrm{x}}_k|}\rangle$ for which we have $S'_i \in S_i$ for every i. There exists some tuple $\bar{\mathsf{c}}_k \in \mathsf{A}^{|\bar{\mathrm{u}}_k|}$ such that $\underline{\lambda}_{k,k}(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \bar{\mathsf{a}}_k, \bar{\mathsf{c}}_k) = T$.

Proof: Recall that for every x_i in $\bar{\mathrm{x}}_k = \langle x_1, \ldots, x_{|\bar{\mathrm{x}}_k|}\rangle$ we have defined the notation $U_{x_i} := \mathrm{vars}(\mathrm{At}_{x_i}) \cap \bar{\mathrm{u}}$. By definition of the fingerprint functions $\lambda_{x,0}$, for every x_i the fact that $S'_i \in \lambda_{x_i,0}(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \bar{\mathsf{a}}_k)$ entails the existence of some tuple $\bar{\mathsf{c}}_{x_i} \in \mathsf{A}^{|\bar{\mathrm{u}}_k|}$ such that $\lambda_{x_i,k}(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \bar{\mathsf{a}}_k, \bar{\mathsf{c}}_{x_i}) = S'_i$. Since we have $U_{x_i} \cap U_{x_j} = \emptyset$ for all $i \neq j$, we can merge the tuples $\bar{\mathsf{c}}_{x_1}, \ldots, \bar{\mathsf{c}}_{x_{\bar{\mathrm{x}}_k}}$ into one tuple $\bar{\mathsf{c}}_k$ of length $|\bar{\mathrm{u}}_k|$ in such a way that $\lambda_{x_i,k}(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \bar{\mathsf{a}}_k, \bar{\mathsf{c}}_k) = S'_i$ holds for every x_i. Then, $\bar{\mathsf{c}}_k$ is the sought tuple. ◊

Claim II: Let k and ℓ be positive integers with $1 \leq k \leq \ell \leq n-1$. Let $\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \bar{\mathsf{a}}_k, \bar{\mathsf{c}}_k, \ldots, \bar{\mathsf{c}}_\ell$ be tuples with $\bar{\mathsf{b}}_i \in \mathsf{A}^{|\bar{\mathrm{y}}_i|}$ for every i, $\bar{\mathsf{a}}_k \in \mathsf{A}^{|\bar{\mathrm{x}}_k|}$, and $\bar{\mathsf{c}}_j \in \mathsf{A}^{|\bar{\mathrm{u}}_j|}$ for every j. Let $\langle S_1, \ldots, S_{|\bar{\mathrm{x}}_k|}\rangle := \underline{\lambda}_{k,\ell}(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \bar{\mathsf{a}}_k, \bar{\mathsf{c}}_k, \ldots, \bar{\mathsf{c}}_\ell)$. Consider any fingerprint $T = \langle S'_1, \ldots, S'_{|\bar{\mathrm{x}}_k|}\rangle$ for which we have $S'_i \in S_i$ for every i. There exists some tuple $\bar{\mathsf{c}}_{\ell+1}$ such that $\underline{\lambda}_{k,\ell+1}(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{k-1}, \bar{\mathsf{a}}_k, \bar{\mathsf{c}}_k, \ldots, \bar{\mathsf{c}}_\ell, \bar{\mathsf{c}}_{\ell+1}) = T$.

Proof: Analogous to the proof of Claim I. ◊

For $k = 1, \ldots, n$ we define $\underline{\mathsf{A}}_k$ as abbreviation of $\mathsf{A}^{|\bar{\mathrm{x}}_1|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{x}}_k|}$. We inductively construct certain representatives $\alpha_{\langle T_1,\ldots,T_k\rangle} \in \underline{\mathsf{A}}_k$ as follows. $\underline{\mathsf{A}}_k$ $\alpha_{\langle\ldots\rangle}$

Let $k = 1$. We partition $\underline{\mathsf{A}}_1$ into sets $\underline{\mathsf{A}}_{1,\langle T_1\rangle}$ with $T_1 \in \mathrm{im}_\sigma(\underline{\lambda}_{1,0})$ by setting $\underline{\mathsf{A}}_{1,\langle T_1\rangle} := \big\{\bar{\mathsf{a}}_1 \in \mathsf{A}^{|\bar{\mathrm{x}}_1|} \;\big|\; \underline{\lambda}_{1,0}(\bar{\mathsf{a}}_1) = T_1\big\}$. We pick one representative $\alpha_{\langle T_1\rangle} \in \underline{\mathsf{A}}_{1,\langle T_1\rangle}$ from each nonempty part $\underline{\mathsf{A}}_{1,\langle T_1\rangle}$. $\underline{\mathsf{A}}_{i,\langle\ldots\rangle}$

Let $k > 1$. We construct subsets $\underline{\mathsf{A}}_{k,\langle T_1,\ldots,T_k\rangle} \subseteq \underline{\mathsf{A}}_k$ with $T_1 \in \mathrm{im}_\sigma(\underline{\lambda}_{1,0}), \ldots, T_k \in \mathrm{im}_\sigma(\underline{\lambda}_{k,0})$ by setting $\underline{\mathsf{A}}_{k,\langle T_1,\ldots,T_k\rangle} :=$

$$\Big\{\big\langle \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{a}}_k\big\rangle \;\Big|\; \begin{array}{l} \bar{\mathsf{a}}_k \in \mathsf{A}^{|\bar{\mathrm{x}}_k|} \text{ and there is some } \alpha_{\langle T_1,\ldots,T_{k-1}\rangle} = \big\langle \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}\big\rangle \\ \quad \text{with } \bar{\mathsf{c}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}_i|} \text{ for every } i \text{ such that} \\ \quad \underline{\lambda}_{1,0}\big(\bar{\mathsf{c}}_1\big) = T_1, \\ \quad \underline{\lambda}_{2,0}\big(\sigma^1_1(\bar{\mathsf{c}}_1), \bar{\mathsf{c}}_2\big) = T_2, \\ \qquad \vdots \\ \quad \underline{\lambda}_{k-1,0}\big(\sigma^1_1(\bar{\mathsf{c}}_1), \ldots, \sigma^1_{k-2}(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-2}), \bar{\mathsf{c}}_{k-1}\big) = T_{k-1}, \\ \quad \underline{\lambda}_{k,0}\big(\sigma^1_1(\bar{\mathsf{c}}_1), \ldots, \sigma^1_{k-1}(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}), \bar{\mathsf{a}}_k\big) = T_k\Big\}\ . \end{array}$$

We pick one representative $\alpha_{\langle T_1,\ldots,T_k\rangle}$ from each nonempty $\underline{\mathsf{A}}_{k,\langle T_1,\ldots,T_k\rangle}$.

Having all the representatives $\alpha_{\langle T_1,\ldots,T_k\rangle}$ at hand, we inductively construct τ, starting from τ^1_1, τ^2_1 and going to τ^1_n, τ^2_n, and show that τ is λ-semi-uniform.

Let $k = 1$. For every $\bar{\mathsf{a}}_1 \in \mathsf{A}^{|\bar{\mathrm{x}}_1|}$ we set $\tau^1_1(\bar{\mathsf{a}}_1) := \sigma^1_1(\alpha_{\langle T\rangle})$, where $T := \underline{\lambda}_{1,0}\big(\bar{\mathsf{a}}_1\big)$. Let $T' := \underline{\lambda}_{1,1}\big(\alpha_{\langle T\rangle}, \sigma^2_1(\alpha_{\langle T\rangle})\big)$. Hence, there are fingerprints $S_1, \ldots, S_{|\bar{\mathrm{x}}_1|}, S'_1, \ldots, S'_{|\bar{\mathrm{x}}_1|}$ such that $T = \langle S_1, \ldots, S_{|\bar{\mathrm{x}}_1|}\rangle$, $T' = \langle S'_1, \ldots, S'_{|\bar{\mathrm{x}}_1|}\rangle$, and $S'_i \in S_i$ for every i. Since $\underline{\lambda}_{1,0}\big(\bar{\mathsf{a}}_1\big) = T$, Claim I entails that there must be some $\bar{\mathsf{b}}_1 \in \mathsf{A}^{|\bar{\mathrm{u}}_1|}$ such that $\underline{\lambda}_{1,1}\big(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1\big) = T'$. We set $\tau^2_1(\bar{\mathsf{a}}_1) := \bar{\mathsf{b}}_1$. In case of $\bar{\mathsf{a}}_1 = \alpha_{\langle T\rangle}$, we make sure that $\bar{\mathsf{b}}_1 = \sigma^2_1(\alpha_{\langle T\rangle})$, i.e. we then set $\tau^2_1(\alpha_{\langle T\rangle}) := \sigma^2_1(\alpha_{\langle T\rangle})$. By construction, τ is λ-semi-uniform up to this point.

Let $k > 1$. Given tuples $\bar{\mathsf{a}}_1 \in \mathsf{A}^{|\bar{\mathrm{x}}_1|}, \ldots, \bar{\mathsf{a}}_k \in \mathsf{A}^{|\bar{\mathrm{x}}_k|}$, let

$$\begin{aligned} T_1 &:= \underline{\lambda}_{1,0}\big(\bar{\mathsf{a}}_1\big)\ , \\ T_2 &:= \underline{\lambda}_{2,0}\big(\tau^1_1(\bar{\mathsf{a}}_1), \bar{\mathsf{a}}_2\big)\ , \\ &\vdots \\ T_k &:= \underline{\lambda}_{k,0}\big(\tau^1_1(\bar{\mathsf{a}}_1), \ldots, \tau^1_{k-1}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{k-1}), \bar{\mathsf{a}}_k\big)\ . \end{aligned}$$

Based on these fingerprints, let $\langle \bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_k \rangle := \alpha_{\langle T_1, \dots, T_k \rangle}$ with $\bar{\mathsf{c}}_i \in \mathsf{A}^{|\bar{\mathsf{x}}_i|}$ for every i. (We argue in Claim IV that such a representative $\alpha_{\langle T_1, \dots, T_k \rangle}$ always exists.) We set $\tau_k^1(\bar{\mathsf{a}}_1, \dots, \bar{\mathsf{a}}_k) := \sigma_k^1(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_k)$. Hence, Condition (a) of the definition of λ-semi-uniformity is satisfied for τ. Moreover, we set $\tau_k^2(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_k) := \sigma_k^2(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_k)$. In order to define $\tau_k^2(\bar{\mathsf{a}}_1, \dots, \bar{\mathsf{a}}_k)$ in cases where $\langle \bar{\mathsf{a}}_1, \dots, \bar{\mathsf{a}}_n \rangle \neq \langle \bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_n \rangle$, we proceed as follows.

Let

$$\begin{aligned}
T_1' &:= \underline{\lambda}_{1,k}\big(\bar{\mathsf{c}}_1, \sigma_1^2(\bar{\mathsf{c}}_1), \dots, \sigma_k^2(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_k)\big) \ , \\
T_2' &:= \underline{\lambda}_{2,k}\big(\tau_1^1(\bar{\mathsf{c}}_1), \bar{\mathsf{c}}_2, \sigma_2^2(\bar{\mathsf{c}}_1, \bar{\mathsf{c}}_2), \dots, \sigma_k^2(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_k)\big) \ , \\
&\ \ \vdots \\
T_k' &:= \underline{\lambda}_{k,k}\big(\tau_1^1(\bar{\mathsf{c}}_1), \dots, \tau_{k-1}^1(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_{k-1}), \bar{\mathsf{c}}_k\big) \ .
\end{aligned}$$

By induction, i.e. by λ-semi-uniformity of τ for $\tau_1^2, \dots, \tau_{k-1}^2$ and by definition of $\tau_1^2, \dots, \tau_{k-1}^2$, we observe that

$$\begin{aligned}
T_j'' &:= \underline{\lambda}_{j,k-1}\big(\tau_1^1(\bar{\mathsf{a}}_1), \dots, \tau_{j-1}^1(\bar{\mathsf{a}}_1, \dots, \bar{\mathsf{a}}_{j-1}), \bar{\mathsf{a}}_j, \tau_j^2(\bar{\mathsf{a}}_1, \dots, \bar{\mathsf{a}}_j), \dots, \tau_{k-1}^2(\bar{\mathsf{a}}_1, \dots, \bar{\mathsf{a}}_{k-1})\big) \\
&= \underline{\lambda}_{j,k-1}\big(\tau_1^1(\bar{\mathsf{c}}_1), \dots, \tau_{j-1}^1(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_{j-1}), \bar{\mathsf{c}}_j, \tau_j^2(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_j), \dots, \tau_{k-1}^2(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_{k-1})\big) \\
&= \underline{\lambda}_{j,k-1}\big(\tau_1^1(\bar{\mathsf{c}}_1), \dots, \tau_{j-1}^1(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_{j-1}), \bar{\mathsf{c}}_j, \sigma_j^2(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_j), \dots, \sigma_{k-1}^2(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_{k-1})\big)
\end{aligned}$$

for every $j = 1, \dots, k-1$, and we have $T_j'' = \langle S_1'', \dots, S_{|\bar{\mathsf{x}}_j|}'' \rangle$ and $T_j' = \langle S_1', \dots, S_{|\bar{\mathsf{x}}_j|}' \rangle$ with $S_i' \in S_i''$ for every i. By virtue of Claims I and II, there exist $\bar{\mathsf{b}}_k^1, \dots, \bar{\mathsf{b}}_k^k \in \mathsf{A}^{|\bar{\mathsf{u}}_k|}$ such that

$$\begin{aligned}
&\underline{\lambda}_{1,k}\big(\bar{\mathsf{a}}_1, \tau_1^2(\bar{\mathsf{a}}_1), \dots, \tau_{k-1}^2(\bar{\mathsf{a}}_1, \dots, \bar{\mathsf{a}}_{k-1}), \bar{\mathsf{b}}_k^1\big) = T_1' \ , \\
&\underline{\lambda}_{2,k}\big(\tau_1^1(\bar{\mathsf{a}}_1), \bar{\mathsf{a}}_2, \tau_2^2(\bar{\mathsf{a}}_1, \bar{\mathsf{a}}_2), \dots, \tau_{k-1}^2(\bar{\mathsf{a}}_1, \dots, \bar{\mathsf{a}}_{k-1}), \bar{\mathsf{b}}_k^2\big) = T_2' \ , \\
&\ \ \vdots \\
&\underline{\lambda}_{k,k}\big(\tau_1^1(\bar{\mathsf{a}}_1), \dots, \tau_{k-1}^1(\bar{\mathsf{a}}_1, \dots, \bar{\mathsf{a}}_{k-1}), \bar{\mathsf{a}}_k, \bar{\mathsf{b}}_k^k\big) = T_k' \ .
\end{aligned}$$

Since the sets At_x do not share any variables from $\bar{\mathsf{u}}_k$, we can merge the tuples $\bar{\mathsf{b}}_k^1, \dots, \bar{\mathsf{b}}_k^k$ into one $\bar{\mathsf{b}}_k$ such that

$$\begin{aligned}
&\underline{\lambda}_{1,k}\big(\bar{\mathsf{a}}_1, \tau_1^2(\bar{\mathsf{a}}_1), \dots, \tau_{k-1}^2(\bar{\mathsf{a}}_1, \dots, \bar{\mathsf{a}}_{k-1}), \bar{\mathsf{b}}_k\big) = T_1' \ , \\
&\underline{\lambda}_{2,k}\big(\tau_1^1(\bar{\mathsf{a}}_1), \bar{\mathsf{a}}_2, \tau_2^2(\bar{\mathsf{a}}_1, \bar{\mathsf{a}}_2), \dots, \tau_{k-1}^2(\bar{\mathsf{a}}_1, \dots, \bar{\mathsf{a}}_{k-1}), \bar{\mathsf{b}}_k\big) = T_2' \ , \\
&\ \ \vdots \\
&\underline{\lambda}_{k,k}\big(\tau_1^1(\bar{\mathsf{a}}_1), \dots, \tau_{k-1}^1(\bar{\mathsf{a}}_1, \dots, \bar{\mathsf{a}}_{k-1}), \bar{\mathsf{a}}_k, \bar{\mathsf{b}}_k\big) = T_k' \ .
\end{aligned}$$

We set $\tau_k^2(\bar{\mathsf{a}}_1, \dots, \bar{\mathsf{a}}_k) := \bar{\mathsf{b}}_k$. Then, τ satisfies Condition (b) of the definition of λ-semi-uniformity.

Notice that, by construction of τ, every representative $\alpha_{\langle T_1, \dots, T_i \rangle} =: \langle \bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_i \rangle$ satisfies

$$\begin{aligned}
&\underline{\lambda}_{k,\ell}\big(\tau_1^1(\bar{\mathsf{c}}_1), \dots, \tau_1^1(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_{k-1}), \bar{\mathsf{c}}_k, \tau_k^2(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_k), \dots, \tau_\ell^2(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_\ell)\big) \\
&= \underline{\lambda}_{k,\ell}\big(\sigma_1^1(\bar{\mathsf{c}}_1), \dots, \sigma_1^1(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_{k-1}), \bar{\mathsf{c}}_k, \sigma_k^2(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_k), \dots, \sigma_\ell^2(\bar{\mathsf{c}}_1, \dots, \bar{\mathsf{c}}_\ell)\big) \qquad (4.1)
\end{aligned}$$

for all k, ℓ with $1 \leq k \leq \ell \leq i$.

Claim III: For every k, $1 \leq k \leq n$, we have $\mathrm{im}_\tau\big(\underline{\lambda}_{k,0}\big) \subseteq \mathrm{im}_\sigma\big(\underline{\lambda}_{k,0}\big)$.

Proof: Fix some k and let $T \in \mathrm{im}_\tau(\underline{\lambda}_{k,0})$. Then, there are tuples $\bar{\mathsf{a}}_1 \in \mathsf{A}^{|\bar{\mathsf{x}}_1|}, \dots, \bar{\mathsf{a}}_k \in \mathsf{A}^{|\bar{\mathsf{x}}_k|}$ such that $\tau_1^1(\bar{\mathsf{a}}_1), \dots, \tau_{k-1}^1(\bar{\mathsf{a}}_1, \dots, \bar{\mathsf{a}}_{k-1})$ are defined and we have

$$T = \underline{\lambda}_{k,0}\big(\tau^1_1(\bar{\mathsf{a}}_1), \ldots, \tau^1_{k-1}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{k-1}), \bar{\mathsf{a}}_k\big).$$

By definition of τ, there is some $\alpha_{\langle T_1,\ldots,T_{k-1}\rangle} = \langle \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}\rangle$ for which we observe

$$\begin{aligned} \tau^1_1(\bar{\mathsf{a}}_1) &= \sigma^1_1(\bar{\mathsf{c}}_1) \\ &\vdots \\ \tau^1_{k-1}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{k-1}) &= \sigma^1_{k-1}(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}) \ . \end{aligned}$$

Consequently, $T = \underline{\lambda}_{k,0}\big(\sigma^1_1(\bar{\mathsf{c}}_1), \ldots, \sigma^1_{k-1}(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}), \bar{\mathsf{a}}_k\big) \in \mathrm{im}_\sigma\big(\underline{\lambda}_{k,0}\big)$. ◊

Claim IV: For every k, $1 \leq k \leq n$, and all tuples $\bar{\mathsf{a}}_1 \in \mathsf{A}^{|\bar{\mathsf{x}}_1|}, \ldots, \bar{\mathsf{a}}_k \in \mathsf{A}^{|\bar{\mathsf{x}}_k|}$ there is a representative $\alpha_{\langle T_1,\ldots,T_k\rangle}$ such that

$$\begin{aligned} T_1 &= \underline{\lambda}_{1,0}\big(\bar{\mathsf{a}}_1\big) \ , \\ T_2 &= \underline{\lambda}_{2,0}\big(\tau^1_1(\bar{\mathsf{a}}_1), \bar{\mathsf{a}}_2\big) \ , \\ &\vdots \\ T_{k-1} &= \underline{\lambda}_{k-1,0}\big(\tau^1_1(\bar{\mathsf{a}}_1), \ldots, \tau^1_{k-2}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{k-2}), \bar{\mathsf{a}}_{k-1}\big) \ , \\ T_k &= \underline{\lambda}_{k,0}\big(\tau^1_1(\bar{\mathsf{a}}_1), \ldots, \tau^1_{k-1}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{k-1}), \bar{\mathsf{a}}_k\big) \ . \end{aligned}$$

Proof: We proceed by induction on k.

Let $k = 1$. Consider any tuple $\bar{\mathsf{a}}_1 \in \mathsf{A}^{|\bar{\mathsf{x}}_1|}$ and set $T_1 := \underline{\lambda}_{1,0}\big(\bar{\mathsf{a}}_1\big)$. Then, $T_1 \in \mathrm{im}_\sigma(\underline{\lambda}_{1,0})$ and we thus have defined the partition $\underline{\mathsf{A}}_{1,\langle T_1\rangle}$. Since $\bar{\mathsf{a}}_1 \in \underline{\mathsf{A}}_{1,\langle T_1\rangle}$, the set is nonempty and there is some representative $\alpha_{\langle T_1\rangle} \in \underline{\mathsf{A}}_{1,\langle T_1\rangle}$.

Let $k > 1$. Consider any sequence of tuples $\bar{\mathsf{a}}_1 \in \mathsf{A}^{|\bar{\mathsf{x}}_1|}, \ldots, \bar{\mathsf{a}}_k \in \mathsf{A}^{|\bar{\mathsf{x}}_k|}$. By Claim III, we have

$$\begin{aligned} T_1 &\in \mathrm{im}_\tau(\underline{\lambda}_{1,0}) \subseteq \mathrm{im}_\sigma(\underline{\lambda}_{1,0}) \ , \\ &\vdots \\ T_k &\in \mathrm{im}_\tau(\underline{\lambda}_{k,0}) \subseteq \mathrm{im}_\sigma(\underline{\lambda}_{k,0}) \ , \end{aligned}$$

and, therefore, we have constructed the subset $\underline{\mathsf{A}}_{k,\langle T_1,\ldots,T_k\rangle} \subseteq \underline{\mathsf{A}}_k$ when defining representatives. It remains to show that this set is not empty.

By induction, there is a unique representative $\alpha_{\langle T_1,\ldots,T_{k-1}\rangle} =: \langle \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}\rangle$ with

$$\begin{aligned} T_1 &= \underline{\lambda}_{1,0}\big(\bar{\mathsf{c}}_1\big) \ , \\ T_2 &= \underline{\lambda}_{2,0}\big(\tau^1_1(\bar{\mathsf{c}}_1), \bar{\mathsf{c}}_2\big) = \underline{\lambda}_{2,0}\big(\sigma^1_1(\bar{\mathsf{c}}_1), \bar{\mathsf{c}}_2\big) \ , \\ &\vdots \\ T_{k-1} &= \underline{\lambda}_{k-1,0}\big(\tau^1_1(\bar{\mathsf{c}}_1), \ldots, \tau^1_{k-2}(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-2}), \bar{\mathsf{c}}_{k-1}\big) \\ &= \underline{\lambda}_{k-1,0}\big(\sigma^1_1(\bar{\mathsf{c}}_1), \ldots, \sigma^1_{k-2}(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-2}), \bar{\mathsf{c}}_{k-1}\big) \ . \end{aligned}$$

This entails $\tau^1_i(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_i) = \tau^1_i(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_i)$ for every i with $1 \leq i \leq k-1$. Therefore,

$$\begin{aligned} T_k &= \underline{\lambda}_{k,0}\big(\tau^1_1(\bar{\mathsf{a}}_1), \ldots, \tau^1_{k-1}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{k-1}), \bar{\mathsf{a}}_k\big) \\ &= \underline{\lambda}_{k,0}\big(\tau^1_1(\bar{\mathsf{c}}_1), \ldots, \tau^1_{k-1}(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}), \bar{\mathsf{a}}_k\big) \\ &= \underline{\lambda}_{k,0}\big(\sigma^1_1(\bar{\mathsf{c}}_1), \ldots, \sigma^1_{k-1}(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}), \bar{\mathsf{a}}_k\big) \ . \end{aligned}$$

Hence, we have $\langle \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{a}}_k\rangle \in \underline{\mathsf{A}}_{k,\langle T_1,\ldots,T_k\rangle}$. Since $\underline{\mathsf{A}}_{k,\langle T_1,\ldots,T_k\rangle}$ contains at least one element, there exists a representative $\alpha_{\langle T_1,\ldots,T_k\rangle}$. ◊

S, S_0, S_x

At this point we have finished showing that the constructed strategy τ is well defined and λ-semi-uniform. It remains to prove $\mathrm{Out}_\tau \subseteq \mathrm{Out}_\sigma$. Let $S \in \mathrm{Out}_\tau$, i.e. there exist tuples $\bar{\mathsf{a}}_1 \in \mathsf{A}^{|\bar{\mathrm{x}}_1|}, \ldots, \bar{\mathsf{a}}_n \in \mathsf{A}^{|\bar{\mathrm{x}}_n|}$ such that $S = \mathrm{out}_\tau(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n)$. We partition S into sets $S_0 := S \cap \mathrm{At}_0$ and $S_x := S \cap \mathrm{At}_x$ for every $x \in \bar{\mathrm{x}}$. Doing so, for every k, $1 \leq k \leq n$, and every $x \in \bar{\mathrm{x}}_k$ we obtain the fingerprint

$$S_x = \lambda_{x,n}\big(\tau_1^1(\bar{\mathsf{a}}_1), \ldots, \tau_{k-1}^1(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{k-1}), \mathsf{a}, \tau_k^2(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k), \ldots, \tau_n^2(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n)\big)\ ,$$

where a from $\bar{\mathsf{a}}_k$ corresponds to x in $\bar{\mathrm{x}}_k$. Combining the sets S_x for one $\bar{\mathrm{x}}_k$ into one tuple, we

T_k

construct sets $T_k := \langle S_{x_1}, \ldots, S_{x_{|\bar{\mathrm{x}}_k|}}\rangle$ for every $\bar{\mathrm{x}}_k = \langle x_1, \ldots, x_{|\bar{\mathrm{x}}_k|}\rangle$. Hence,

$$T_k = \underline{\lambda}_{k,n}\big(\tau_1^1(\bar{\mathsf{a}}_1), \ldots, \tau_{k-1}^1(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{k-1}), \bar{\mathsf{a}}_k, \tau_k^2(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k), \ldots, \tau_n^2(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n)\big)$$

for every $k = 1, \ldots, n$.

T'_k

Let $T'_k = \underline{\lambda}_{k,0}\big(\tau_1^1(\bar{\mathsf{a}}_1), \ldots, \tau_{k-1}^1(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{k-1}), \bar{\mathsf{a}}_k\big)$ for $k = 1, \ldots, n$. By virtue of Claim IV, there is some representative $\alpha_{\langle T'_1, \ldots, T'_n\rangle} = \langle \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_n\rangle$. Because of λ-semi-uniformity of τ and due to Equation (4.1), we have

$$\begin{aligned}
&\underline{\lambda}_{k,\ell}\big(\tau_1^1(\bar{\mathsf{a}}_1), \ldots, \tau_{k-1}^1(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{k-1}), \bar{\mathsf{a}}_k, \tau_k^2(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k), \ldots, \tau_\ell^2(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_\ell)\big)\\
&= \underline{\lambda}_{k,\ell}\big(\tau_1^1(\bar{\mathsf{c}}_1), \ldots, \tau_{k-1}^1(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}), \bar{\mathsf{c}}_k, \tau_k^2(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_k), \ldots, \tau_\ell^2(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_\ell)\big)\\
&= \underline{\lambda}_{k,\ell}\big(\sigma_1^1(\bar{\mathsf{c}}_1), \ldots, \sigma_{k-1}^1(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}), \bar{\mathsf{c}}_k, \sigma_k^2(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_k), \ldots, \sigma_\ell^2(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_\ell)\big)
\end{aligned}$$

for all k, ℓ with $1 \leq k \leq \ell \leq n$. Consequently, for every k we get

$$\underline{\lambda}_{k,n}\big(\sigma_1^1(\bar{\mathsf{c}}_1), \ldots, \sigma_{k-1}^1(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}), \bar{\mathsf{c}}_k, \sigma_k^2(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_k), \ldots, \sigma_n^2(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_n)\ \big) = T_k.$$

When we decompose T_k into its constituents $S_{x_1}, \ldots, S_{x_{|\bar{\mathrm{x}}_k|}}$, we get for every $x \in \bar{\mathrm{x}}_k$ that

$$S_x = \lambda_{x,n}\big(\sigma_1^1(\bar{\mathsf{c}}_1), \ldots, \sigma_{k-1}^1(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}), \mathsf{c}, \sigma_k^2(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_k), \ldots, \sigma_n^2(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_n)\ \big),$$

where c from $\bar{\mathsf{c}}_k$ corresponds to x in $\bar{\mathrm{x}}_k$. Since the union over the S_x yields S, this entails $S = \mathrm{out}_\sigma(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_n) \in \mathrm{Out}_\sigma$. Altogether, we thus have shown $\mathrm{Out}_\tau \subseteq \mathrm{Out}_\sigma$. □

Consider a λ-semi-uniform strategy σ under $\mathcal{A}$. Since σ does not necessarily have a finite image, we cannot expect that the substructure $\mathcal{A}|_\sigma$ of $\mathcal{A}$ is finite. However, starting from $\mathcal{A}$ we can construct another, possibly infinite structure $\mathcal{B}$ accompanied with a satisfying λ-semi-uniform strategy τ with a finite image. Then, $\mathcal{B}|_\tau$ is a finite model for φ.

Theorem 4.3.5. *Every satisfiable relational GAF sentence $\varphi := \forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1\bar{\mathrm{u}}_1 \ldots \forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n\bar{\mathrm{u}}_n.\,\psi$ in standard form with quantifier-free ψ has a model with at most $\big(p(\mathrm{len}(\varphi))\big)^{2|\bar{\mathrm{x}}|} \cdot \big(2^{\uparrow n+1}(|\mathrm{At}|)\big)^{2|\bar{\mathrm{x}}|+2}$ elements, where p is some polynomial, $\bar{\mathrm{x}} := \bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_n$, $\bar{\mathrm{y}} := \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_n$ and where we assume $|\bar{\mathrm{x}}| \geq 1$ and $|\bar{\mathrm{y}}| \geq 1$.*

The theorem is a consequence of the following lemma.

Lemma 4.3.6. *Suppose that $\mathcal{A}$ is a model of φ. Let σ be a λ-semi-uniform strategy that is satisfying for φ. Then, φ has a finite model $\mathcal{C}$ with at most $\big(p(\mathrm{len}(\varphi))\big)^{2|\bar{\mathrm{x}}|} \cdot \big(2^{\uparrow n+1}(|\mathrm{At}|)\big)^{2|\bar{\mathrm{x}}|+2}$ domain elements, where p is some polynomial and where we assume $|\bar{\mathrm{x}}| \geq 1$ and $|\bar{\mathrm{y}}| \geq 1$.*

$\underline{\mathsf{A}}_k$
$\mathcal{T}_{\sigma^1}$

Proof. We take over the definition of the sets $\underline{\mathsf{A}}_k$ from the proof of Lemma 4.3.4, based on $\mathcal{A}$'s domain. Let $\mathcal{T}_{\sigma^1}$ be the *target set* of the mappings σ_k^1, which we define by

$$\mathcal{T}_{\sigma^1} := \big\{\mathsf{b} \in \mathsf{A} \mid \sigma_k^1(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k) = \langle \ldots \mathsf{b} \ldots\rangle \text{ for some } k \text{ and some } \bar{\mathsf{a}}_1 \in \mathsf{A}^{|\bar{\mathrm{x}}_1|}, \ldots, \bar{\mathsf{a}}_k \in \mathsf{A}^{|\bar{\mathrm{x}}_k|}\big\}.$$

Notice that, since σ is λ-semi-uniform, $\mathcal{T}_{\sigma^1}$ is finite.

$\sim$

We start by defining an equivalence relation $\sim$ on tuples taken from the sets $\underline{\mathsf{A}}_k$. Let $\langle \bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k\rangle$ and $\langle \bar{\mathsf{a}}'_1, \ldots, \bar{\mathsf{a}}'_k\rangle$ be tuples in $\underline{\mathsf{A}}_k$ for some k, $1 \leq k \leq n$. We say $\langle \bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k\rangle \sim \langle \bar{\mathsf{a}}'_1, \ldots, \bar{\mathsf{a}}'_k\rangle$ if and only if the following conditions are satisfied:

(1) For all indices i, j we have $\mathsf{a}_{i,j} \in \mathcal{T}_{\sigma^1}$ if and only if $\mathsf{a}'_{i,j} \in \mathcal{T}_{\sigma^1}$. Moreover, if $\mathsf{a}_{i,j}, \mathsf{a}'_{i,j} \in \mathcal{T}_{\sigma^1}$ then $\mathsf{a}_{i,j} = \mathsf{a}'_{i,j}$.

(2) For every i, $1 \leq i \leq k$, we have
$$\underline{\lambda}_{i,0}\big(\sigma_1^1(\bar{\mathsf{a}}_1),\ldots,\sigma_{i-1}^1(\bar{\mathsf{a}}_1,\ldots,\bar{\mathsf{a}}_{i-1}),\bar{\mathsf{a}}_i\big) \;=\; \underline{\lambda}_{i,0}\big(\sigma_1^1(\bar{\mathsf{a}}'_1),\ldots,\sigma_{i-1}^1(\bar{\mathsf{a}}'_1,\ldots,\bar{\mathsf{a}}'_{i-1}),\bar{\mathsf{a}}'_i\big).$$

Since $\mathcal{T}_{\sigma^1}$ is finite and because there are only finitely many fingerprints, the relation $\sim$ induces finitely many equivalence classes that partition the set $\underline{\mathsf{A}}_1 \cup \ldots \cup \underline{\mathsf{A}}_n$. For each nonempty equivalence class $[\langle \bar{\mathsf{a}}_1,\ldots,\bar{\mathsf{a}}_k\rangle]_\sim$ we fix some representative $\alpha_{[\langle \bar{\mathsf{a}}_1,\ldots,\bar{\mathsf{a}}_k\rangle]_\sim}$ such that there are tuples $\bar{\mathsf{c}}_1,\ldots,\bar{\mathsf{c}}_k$ with $\bar{\mathsf{c}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}_i|}$ and $\langle\bar{\mathsf{c}}_1\rangle = \alpha_{[\langle \bar{\mathsf{a}}_1\rangle]_\sim}, \langle\bar{\mathsf{c}}_1,\bar{\mathsf{c}}_2\rangle = \alpha_{[\langle \bar{\mathsf{a}}_1,\bar{\mathsf{a}}_2\rangle]_\sim},\ldots,\langle\bar{\mathsf{c}}_1,\ldots,\bar{\mathsf{c}}_k\rangle = \alpha_{[\langle \bar{\mathsf{a}}_1,\ldots,\bar{\mathsf{a}}_k\rangle]_\sim}$. The existence of such representatives is a consequence of the following claim, Claim I, and the fact that $\langle\bar{\mathsf{c}}_1,\ldots,\bar{\mathsf{c}}_k\rangle \sim \langle\bar{\mathsf{a}}_1,\ldots,\bar{\mathsf{a}}_k\rangle$ entails $\langle\bar{\mathsf{c}}_1,\ldots,\bar{\mathsf{c}}_i\rangle \sim \langle\bar{\mathsf{a}}_1,\ldots,\bar{\mathsf{a}}_i\rangle$ for every i with $1 \leq i \leq k$.

Claim I: Consider any positive integer ℓ and two tuples $\langle\bar{\mathsf{b}}_1,\ldots,\bar{\mathsf{b}}_\ell\rangle \sim \langle\bar{\mathsf{d}}_1,\ldots,\bar{\mathsf{d}}_\ell\rangle$ with $\bar{\mathsf{b}}_i,\bar{\mathsf{d}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}_i|}$. For every k with $\ell \leq k \leq n$ and all $\bar{\mathsf{d}}_{\ell+1},\ldots,\bar{\mathsf{d}}_k$ with $\bar{\mathsf{d}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}_i|}$ we have $\langle\bar{\mathsf{b}}_1,\ldots,\bar{\mathsf{b}}_\ell,\bar{\mathsf{d}}_{\ell+1},\ldots,\bar{\mathsf{d}}_k\rangle \sim \langle\bar{\mathsf{d}}_1,\ldots,\bar{\mathsf{d}}_k\rangle$.

Proof: As Condition (1) of the definition of $\sim$ is obviously satisfied, we concentrate on Condition (2). Our assumption $\langle\bar{\mathsf{b}}_1,\ldots,\bar{\mathsf{b}}_\ell\rangle \sim \langle\bar{\mathsf{d}}_1,\ldots,\bar{\mathsf{d}}_\ell\rangle$ entails

$$\begin{aligned}
\underline{\lambda}_{1,0}\big(\bar{\mathsf{b}}_1\big) &= \underline{\lambda}_{1,0}\big(\bar{\mathsf{d}}_1\big)\ ,\\
\underline{\lambda}_{2,0}\big(\sigma_1^1(\bar{\mathsf{b}}_1),\bar{\mathsf{b}}_2\big) &= \underline{\lambda}_{1,0}\big(\sigma_1^1(\bar{\mathsf{d}}_1),\bar{\mathsf{d}}_2\big)\ ,\\
&\vdots\\
\underline{\lambda}_{\ell,0}\big(\sigma_1^1(\bar{\mathsf{b}}_1),\ldots,\sigma_{\ell-1}^1(\bar{\mathsf{b}}_1,\ldots,\bar{\mathsf{b}}_{\ell-1}),\bar{\mathsf{b}}_\ell\big) &= \underline{\lambda}_{1,0}\big(\sigma_1^1(\bar{\mathsf{d}}_1),\ldots,\sigma_{\ell-1}^1(\bar{\mathsf{d}}_1,\ldots,\bar{\mathsf{d}}_{\ell-1}),\bar{\mathsf{d}}_\ell\big)\ .
\end{aligned}$$

Hence, λ-semi-uniformity of σ leads to $\sigma_1^1(\bar{\mathsf{b}}_1) = \sigma_1^1(\bar{\mathsf{d}}_1)$, …, $\sigma_\ell^1(\bar{\mathsf{b}}_1,\ldots,\bar{\mathsf{b}}_\ell) = \sigma_\ell^1(\bar{\mathsf{d}}_1,\ldots,\bar{\mathsf{d}}_\ell)$. We next show for every i, $\ell \leq i \leq k$, that
(a) $\underline{\lambda}_{i,0}\big(\sigma_1^1(\bar{\mathsf{b}}_1),\ldots,\sigma_\ell^1(\bar{\mathsf{b}}_1,\ldots,\bar{\mathsf{b}}_\ell),\sigma_{\ell+1}^1(\bar{\mathsf{b}}_1,\ldots,\bar{\mathsf{b}}_\ell,\bar{\mathsf{d}}_{\ell+1}),\ldots,$
$\sigma_{i-1}^1(\bar{\mathsf{b}}_1,\ldots,\bar{\mathsf{b}}_\ell,\bar{\mathsf{d}}_{\ell+1},\ldots,\bar{\mathsf{d}}_{i-1}),\bar{\mathsf{d}}_i\big)$
$= \underline{\lambda}_{i,0}\big(\sigma_1^1(\bar{\mathsf{d}}_1),\ldots,\sigma_{i-1}^1(\bar{\mathsf{d}}_1,\ldots,\bar{\mathsf{d}}_{i-1}),\bar{\mathsf{d}}_i\big)$
and
(b) $\sigma_i^1(\bar{\mathsf{b}}_1,\ldots,\bar{\mathsf{b}}_\ell,\bar{\mathsf{d}}_{\ell+1},\ldots,\bar{\mathsf{d}}_i) = \sigma_i^1(\bar{\mathsf{d}}_1,\ldots,\bar{\mathsf{d}}_i)$.
We proceed by induction on i. For the base case $i = \ell$ there is nothing to do, as the above observations already state what we have to show. Consider the case $i \geq \ell+1$. By induction, Equation (b) for $\ell,\ldots,i-1$ entails
$$\begin{aligned}
&\underline{\lambda}_{i,0}\big(\sigma_1^1(\bar{\mathsf{d}}_1),\ldots,\sigma_{i-1}^1(\bar{\mathsf{d}}_1,\ldots,\bar{\mathsf{d}}_{i-1}),\bar{\mathsf{d}}_i\big)\\
&= \underline{\lambda}_{i,0}\big(\sigma_1^1(\bar{\mathsf{b}}_1),\ldots,\sigma_{i-1}^1(\bar{\mathsf{b}}_1,\ldots,\bar{\mathsf{b}}_\ell,\bar{\mathsf{d}}_{\ell+1},\ldots,\bar{\mathsf{d}}_{i-1}),\bar{\mathsf{d}}_i\big).
\end{aligned}$$
By λ-semi-uniformity, this together with the previous observations and (a) for $\ell,\ldots,i-1$ implies $\sigma_i^1(\bar{\mathsf{b}}_1,\ldots,\bar{\mathsf{b}}_\ell,\bar{\mathsf{d}}_{\ell+1},\ldots,\bar{\mathsf{d}}_i) = \sigma_i^1(\bar{\mathsf{d}}_1,\ldots,\bar{\mathsf{d}}_i)$.

This finishes the inductive proof of (a) and (b). Now, part (a) for $i = k$ immediately entails that Condition (2) for $\langle\bar{\mathsf{b}}_1,\ldots,\bar{\mathsf{b}}_\ell,\bar{\mathsf{d}}_{\ell+1},\ldots,\bar{\mathsf{d}}_k\rangle \sim \langle\bar{\mathsf{d}}_1,\ldots,\bar{\mathsf{d}}_k\rangle$ is satisfied. ◊

Next, we construct a certain structure $\mathcal{B}$ from $\mathcal{A}$. Let $\uplus$ denote the disjoint-union operator. We compose the domain of $\mathcal{B}$ by setting $\mathsf{B} := \mathsf{D}_{-1} \uplus \mathsf{D}_0 \uplus \mathsf{D}_1 \uplus \mathsf{D}_2$ for four subdomains $\mathsf{D}_{-1},\mathsf{D}_0,\mathsf{D}_1,\mathsf{D}_2$. Every D_i with $i \geq 0$ is a disjoint union of layers: one layer $\mathsf{D}_{i,C,u}$ for every equivalence class C induced by $\sim$ on the set $\underline{\mathsf{A}}_1 \cup \ldots \cup \underline{\mathsf{A}}_n$ and every $u \in \bar{\mathrm{u}}$. Each of these layers $\mathsf{D}_{i,C,u}$ is a copy of $\mathcal{A}$'s domain A. We impose a similar layered structure on D_{-1}, however with references to $y \in \bar{\mathrm{y}}$ instead of $u \in \bar{\mathrm{u}}$. Given any domain element $\mathsf{a} \in \mathsf{D}_{-1} \uplus \mathsf{D}_0 \uplus \mathsf{D}_1 \uplus \mathsf{D}_2$, we denote by $\mathsf{a}^\downarrow$ the element $\mathsf{a}' \in \mathsf{A}$ from which the copy a originated. We extend this notation to tuples $\bar{\mathsf{a}}$ by setting by $\bar{\mathsf{a}}^\downarrow := \langle\mathsf{a}_1^\downarrow,\ldots,\mathsf{a}_{|\bar{\mathsf{a}}|}^\downarrow\rangle$. We further extend the notation to variable assignments, i.e. for every variable assignment β over $\mathcal{B}$'s domain we write $\beta^\downarrow$ to address the variable assignment over $\mathcal{A}$'s domain that is defined by setting $\beta^\downarrow(v) := \big(\beta(v)\big)^\downarrow$ for every variable v.

$\mathsf{D}_{-1},\mathsf{D}_0,\mathsf{D}_1,\mathsf{D}_2$
$\mathsf{D}_{i,C,u}$, $\mathsf{D}_{-1,C,y}$
$\mathsf{a}^\downarrow$, $\bar{\mathsf{a}}^\downarrow$, $\beta^\downarrow$

Based on the strategy $\sigma = \langle\sigma_1,\ldots,\sigma_n\rangle$, we construct a new strategy $\tau := \langle\tau_1,\ldots,\tau_n\rangle$ with $\tau_i : \mathsf{B}^{|\bar{\mathrm{x}}_1|} \times \ldots \times \mathsf{B}^{|\bar{\mathrm{x}}_i|} \to \mathsf{B}^{|\bar{\mathrm{y}}_i|} \times \mathsf{B}^{|\bar{\mathrm{u}}_i|}$ as follows.

$\tau = \langle\tau_1,\ldots,\tau_n\rangle$

Consider any tuple $\bar{\mathsf{a}} \in \mathsf{B}^{|\bar{\mathrm{x}}_1|}$. Let $C := [\bar{\mathsf{a}}^\downarrow]_\sim \subseteq \underline{\mathsf{A}}_1$ be the equivalence class to which $\bar{\mathsf{a}}^\downarrow$ belongs and let $\langle\bar{\mathsf{c}}\rangle := \alpha_C$ be the distinguished representative in C. We define τ_1 in such a way that

(a) $\big(\tau_1^1(\bar{\mathsf{a}})\big)^{\downarrow} = \sigma_1^1(\bar{\mathsf{c}})$ and $\big(\tau_1^2(\bar{\mathsf{a}})\big)^{\downarrow} = \sigma_1^2(\bar{\mathsf{c}})$,

(b) every domain element b in the tuple $\tau_1^1(\bar{\mathsf{a}})$ belongs to $\mathsf{D}_{-1,C,y}$ where y is the variable in $\bar{\mathrm{y}}_1$ to which b corresponds, and

(c) for every $x \in \bar{\mathrm{x}}_1$ we have that if the corresponding $\mathsf{a} \in \bar{\mathsf{a}}$ stems from D_i then every element b in the tuple $\tau_1^2(\bar{\mathsf{a}})$ that corresponds to some variable $u \in U_x \cap \bar{\mathrm{u}}_1$ is taken from $\mathsf{D}_{(i+1 \bmod 3),C,u}$.

Let $k > 1$. Consider any sequence of tuples $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k$ with $\bar{\mathsf{a}}_i \in \mathsf{B}^{|\bar{\mathrm{x}}_i|}$. Let $C := [\langle \bar{\mathsf{a}}_1^{\downarrow}, \ldots, \bar{\mathsf{a}}_k^{\downarrow}\rangle]_{\sim} \subseteq \underline{\mathrm{A}}_k$ be the equivalence class to which $\langle \bar{\mathsf{a}}_1^{\downarrow}, \ldots, \bar{\mathsf{a}}_k^{\downarrow}\rangle$ belongs and let $\langle \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_k\rangle := \alpha_C$ be the distinguished representative in C. We define τ_k in such a way that the following conditions are met:

(a) $\big(\tau_k^1(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k)\big)^{\downarrow} = \sigma_k^1(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_k)$ and $\big(\tau_k^2(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k)\big)^{\downarrow} = \sigma_k^2(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_k)$.

(b) Every domain element b in $\tau_k^1(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k)$ belongs to $\mathsf{D}_{-1,C,y}$, where y is the variable in $\bar{\mathrm{y}}_k$ to which b corresponds.

(c) For every $x \in \bar{\mathrm{x}}_\ell$ with $1 \leq \ell \leq k$ we have that if the corresponding $\mathsf{a} \in \bar{\mathsf{a}}_\ell$ stems from D_i then every element b in the tuple $\tau_k^2(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k)$ that corresponds to some variable $u \in U_x \cap \bar{\mathrm{u}}_k$ is taken from $\mathsf{D}_{(i+1 \bmod 3),C,u}$.

Claim II: For every k we have $\sigma_k^1(\bar{\mathsf{a}}_1^{\downarrow}, \ldots, \bar{\mathsf{a}}_k^{\downarrow}) = \big(\tau_k^1(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k)\big)^{\downarrow}$.

Proof: We prove this claim by induction on k.

For the base case $k = 1$ the claim holds due to our assumption that σ is λ-semi-uniform. More precisely, we have $\big(\tau_1^1(\bar{\mathsf{a}}_1)\big)^{\downarrow} = \sigma_1^1(\bar{\mathsf{c}}_1) = \sigma_1^1(\bar{\mathsf{a}}_1^{\downarrow})$ where $\bar{\mathsf{c}}_1 = \alpha_{[\bar{\mathsf{a}}_1^{\downarrow}]_{\sim}}$.

In the case $k > 1$ let $T_i := \underline{\lambda}_{i,0}\big((\tau_1^1(\bar{\mathsf{a}}_1))^{\downarrow}, \ldots, (\tau_{i-1}^1(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{i-1}))^{\downarrow}, \bar{\mathsf{a}}_i^{\downarrow}\big)$ for every i, $1 \leq i \leq k$. Since we have $\sigma_i^1(\bar{\mathsf{a}}_1^{\downarrow}, \ldots, \bar{\mathsf{a}}_i^{\downarrow}) = \big(\tau_i^1(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_i)\big)^{\downarrow}$ for every $i = 1, \ldots, k-1$ by induction, we get

$$T_i = \underline{\lambda}_{i,0}\big(\sigma_1^1(\bar{\mathsf{a}}_1^{\downarrow}), \ldots, \sigma_{i-1}^1(\bar{\mathsf{a}}_1^{\downarrow}, \ldots, \bar{\mathsf{a}}_{i-1}^{\downarrow}), \bar{\mathsf{a}}_i^{\downarrow}\big)$$

for every i, $1 \leq i \leq k$. Let $\langle \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_k\rangle := \alpha_{[\langle \bar{\mathsf{a}}_1^{\downarrow}, \ldots, \bar{\mathsf{a}}_k^{\downarrow}\rangle]_{\sim}}$. By λ-semi-uniformity of σ and due to the construction of τ, we then have $\sigma_k^1(\bar{\mathsf{a}}_1^{\downarrow}, \ldots, \bar{\mathsf{a}}_k^{\downarrow}) = \sigma_k^1(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_k) = \big(\tau_k^1(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k)\big)^{\downarrow}$. Hence, the claim follows. ◊

We next define how $\mathcal{B}$ interprets predicate symbols. For every predicate symbol P occurring in φ we define $P^{\mathcal{B}}$ to be the smallest set satisfying the following properties for all tuples $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n$ with $\bar{\mathsf{a}}_i \in \mathsf{B}^{|\bar{\mathrm{x}}_i|}$. Let $\langle \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_n\rangle := \alpha_{[\langle \bar{\mathsf{a}}_1^{\downarrow}, \ldots, \bar{\mathsf{a}}_n^{\downarrow}\rangle]_{\sim}}$.

(i) For every atom $A(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n) \in \mathrm{At}_0$ we require $\mathcal{B} \models A\big(\tau_1^1(\bar{\mathsf{a}}_1), \ldots, \tau_n^1(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n)\big)$ if and only if $\mathcal{A} \models A\big(\sigma_1^1(\bar{\mathsf{c}}_1), \ldots, \sigma_n^1(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_n)\big)$.

(ii) For every k, $1 \leq k \leq n$, every $x \in \bar{\mathrm{x}}_k$, and every atom $A(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_{k-1}, x, \bar{\mathrm{u}}_k, \ldots, \bar{\mathrm{u}}_n) \in \mathrm{At}_x$ we require

$$\mathcal{B} \models A\big(\tau_1^1(\bar{\mathsf{a}}_1), \ldots, \tau_{k-1}^1(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{k-1}), \mathsf{a}, \tau_k^2(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k), \ldots, \tau_n^2(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n)\big)$$

if and only if

$$\mathcal{A} \models A\big(\sigma_1^1(\bar{\mathsf{c}}_1), \ldots, \sigma_{k-1}^1(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}), \mathsf{c}, \sigma_k^2(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_k), \ldots, \sigma_n^2(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_n)\big),$$

where a from $\bar{\mathsf{a}}_k$ and c from $\bar{\mathsf{c}}_k$ correspond to $x \in \bar{\mathrm{x}}_k$.

Notice that the last line implies

$$A \in \lambda_{x,n}\big(\sigma_1^1(\bar{\mathsf{c}}_1), \ldots, \sigma_{k-1}^1, \mathsf{c}, \sigma_k^2(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_k), \ldots, \sigma_n^2(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_n)\big)$$

where c corresponds to x in $\bar{\mathrm{x}}_k$.

Claim III: The structure $\mathcal{B}$ is well defined.

Proof: We assume to the contrary that the definition of $\mathcal{B}$ is contradictory while $\mathcal{A}$ is well defined. Consider any two sequences of tuples $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n$ and $\bar{\mathsf{a}}'_1, \ldots, \bar{\mathsf{a}}'_n$ with $\bar{\mathsf{a}}_k, \bar{\mathsf{a}}'_k \in \mathsf{B}^{|\bar{\mathrm{x}}_k|}$ for every k. Let β be the variable assignment that maps every $\bar{\mathrm{x}}_k$ to $\bar{\mathsf{a}}_k$, every $\bar{\mathrm{y}}_k$ to $\tau^1_k(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k)$, and every $\bar{\mathrm{u}}_k$ to $\tau^2_k(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k)$. Let β' be defined analogously based on $\bar{\mathsf{a}}'_1, \ldots, \bar{\mathsf{a}}'_n$. Suppose there are atoms A, B occurring in φ such that $\mathcal{B}, \beta \models A$ and $\mathcal{B}, \beta' \not\models B$. Moreover, suppose that A has the shape $P(s_1, \ldots, s_m)$ and B has the shape $P(t_1, \ldots, t_m)$ and that we have $\beta(s_\ell) = \beta'(t_\ell)$ for every ℓ. Recall that, since φ is relational, all the terms s_ℓ and t_ℓ are in fact variables.

$\bar{\mathsf{a}}_k, \bar{\mathsf{a}}'_k$

β, β'

A, B, s_i, t_i

Let $T_1, \ldots, T_n$ be the unique sequence of fingerprints defined by

T_k

$$\begin{aligned} T_k &:= \underline{\lambda}_{k,0}\big((\tau^1_1(\bar{\mathsf{a}}_1))^{\downarrow}, \ldots, (\tau^1_{k-1}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{k-1}))^{\downarrow}, \bar{\mathsf{a}}^{\downarrow}_k\big) \\ &= \underline{\lambda}_{k,0}\big(\sigma^1_1(\bar{\mathsf{a}}^{\downarrow}_1), \ldots, \sigma^1_{k-1}(\bar{\mathsf{a}}^{\downarrow}_1, \ldots, \bar{\mathsf{a}}^{\downarrow}_{k-1}), \bar{\mathsf{a}}^{\downarrow}_k\big) \end{aligned}$$

for every k. Let $\langle \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_n \rangle := \alpha_{[\langle \bar{\mathsf{a}}^{\downarrow}_1, \ldots, \bar{\mathsf{a}}^{\downarrow}_n \rangle]_\sim}$ and let γ be the variable assignment that maps every $\bar{\mathrm{x}}_k$ to $\bar{\mathsf{c}}_k$, every $\bar{\mathrm{y}}_k$ to $\sigma^1_k(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_k)$, and every $\bar{\mathrm{u}}_k$ to $\sigma^2_k(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_k)$. We define $T'_1, \ldots, T'_n$ and $\bar{\mathsf{c}}'_1, \ldots, \bar{\mathsf{c}}'_n$ and γ' analogously, based on $\bar{\mathsf{a}}'_1, \ldots, \bar{\mathsf{a}}'_n$.

$\langle \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_n \rangle$, $\langle \bar{\mathsf{c}}'_1, \ldots, \bar{\mathsf{c}}'_n \rangle$, T'_k, γ, γ'

By Requirements (i) and (ii), our assumptions entail

$$\mathcal{A}, \gamma \models A \text{ and } \mathcal{A}, \gamma' \not\models B. \tag{4.2}$$

Consider the variable assignments $\beta^{\downarrow}$ and $\beta'^{\downarrow}$. Clearly, for all variables v, v' we have that $\beta(v) = \beta'(v')$ implies $\beta^{\downarrow}(v) = \beta'^{\downarrow}(v')$. Hence, we have

$$\mathcal{A}(\beta^{\downarrow})(A) = P\big(\beta^{\downarrow}(s_1), \ldots, \beta^{\downarrow}(s_m)\big) = P\big(\beta'^{\downarrow}(t_1), \ldots, \beta'^{\downarrow}(t_m)\big) = \mathcal{A}(\beta'^{\downarrow})(B). \tag{4.3}$$

Regarding the atoms A and B, we distinguish several cases.

Case (A1). If $A(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n) \in \mathrm{At}_0$, we observe the following. By definition of γ, $\mathcal{A}, \gamma \models A$ translates to $\mathcal{A} \models A\big(\sigma^1_1(\bar{\mathsf{c}}_1), \ldots, \sigma^1_n(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_n)\big)$. Since τ is defined such that $\sigma^1_i(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_i) = \big(\tau^1_i(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_i)\big)^{\downarrow}$ for every i, we thus obtain

$$\mathcal{A} \models A\big((\tau^1_1(\bar{\mathsf{a}}_1))^{\downarrow}, \ldots, (\tau^1_n(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n))^{\downarrow}\big).$$

By definition of β, this translates to $\mathcal{A}, \beta^{\downarrow} \models A$.

Case (A2). If $A(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_{k-1}, \bar{\mathrm{x}}_k, \bar{\mathrm{u}}_k, \ldots, \bar{\mathrm{u}}_n) \in \mathrm{At}_x$ for some $x \in \bar{\mathrm{x}}_k$, we observe the following. By definition of γ, $\mathcal{A}, \gamma \models A$ translates to

x, k

$$\mathcal{A} \models A\big(\sigma^1_1(\bar{\mathsf{c}}_1), \ldots, \sigma^1_{k-1}(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}), \bar{\mathsf{c}}_k, \sigma^2_k(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_k), \ldots, \sigma^2_n(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_n)\big)\ .$$

Because of $\langle \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_n \rangle \sim \langle \bar{\mathsf{a}}^{\downarrow}_1, \ldots, \bar{\mathsf{a}}^{\downarrow}_n \rangle$, λ-semi-uniformity of σ entails

$$\begin{aligned} A &\in \lambda_{k,n}\big(\sigma^1_1(\bar{\mathsf{c}}_1), \ldots, \sigma^1_{k-1}(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{k-1}), \bar{\mathsf{c}}_k, \sigma^2_k(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_k), \ldots, \sigma^2_n(\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_n)\big) \\ &= \lambda_{k,n}\big(\sigma^1_1(\bar{\mathsf{a}}^{\downarrow}_1), \ldots, \sigma^1_{k-1}(\bar{\mathsf{a}}^{\downarrow}_1, \ldots, \bar{\mathsf{a}}^{\downarrow}_{k-1}), \bar{\mathsf{a}}^{\downarrow}_k, \sigma^2_k(\bar{\mathsf{a}}^{\downarrow}_1, \ldots, \bar{\mathsf{a}}^{\downarrow}_k), \ldots, \sigma^2_n(\bar{\mathsf{a}}^{\downarrow}_1, \ldots, \bar{\mathsf{a}}^{\downarrow}_n)\big)\ . \end{aligned}$$

Therefore, we get

$$\mathcal{A} \models A\big(\sigma^1_1(\bar{\mathsf{a}}^{\downarrow}_1), \ldots, \sigma^1_{k-1}(\bar{\mathsf{a}}^{\downarrow}_1, \ldots, \bar{\mathsf{a}}^{\downarrow}_{k-1}), \bar{\mathsf{a}}^{\downarrow}_k, \sigma^2_k(\bar{\mathsf{a}}^{\downarrow}_1, \ldots, \bar{\mathsf{a}}^{\downarrow}_k), \ldots, \sigma^2_n(\bar{\mathsf{a}}^{\downarrow}_1, \ldots, \bar{\mathsf{a}}^{\downarrow}_n)\big)\ .$$

By virtue of Claim II, this can be rewritten to

$$\mathcal{A} \models A\big((\tau^1_1(\bar{\mathsf{a}}_1))^{\downarrow}, \ldots, (\tau^1_{k-1}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{k-1}))^{\downarrow}, \bar{\mathsf{a}}^{\downarrow}_k, \sigma^2_k(\bar{\mathsf{a}}^{\downarrow}_1, \ldots, \bar{\mathsf{a}}^{\downarrow}_k), \ldots, \sigma^2_n(\bar{\mathsf{a}}^{\downarrow}_1, \ldots, \bar{\mathsf{a}}^{\downarrow}_n)\big)\ .$$

Case (B1). If $B \in \mathrm{At}_0$, we conclude $\mathcal{A}, \beta'^{\downarrow} \not\models B$ in analogy to Case (A1).

x', k' **Case (B2).** If $B(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_{k'-1}, \bar{\mathrm{x}}_{k'}, \bar{\mathrm{u}}_{k'}, \ldots, \bar{\mathrm{u}}_n) \in \mathrm{At}_{x'}$ for some $x' \in \bar{\mathrm{x}}_{k'}$, we can derive

$$\mathcal{A} \not\models B\big((\tau^1_1(\bar{\mathsf{a}}'_1))^{\downarrow}, \ldots, (\tau^1_{k'-1}(\bar{\mathsf{a}}'_1, \ldots, \bar{\mathsf{a}}'_{k'-1}))^{\downarrow}, \bar{\mathsf{a}}'^{\downarrow}_{k'}, \sigma^2_{k'}(\bar{\mathsf{a}}'^{\downarrow}_1, \ldots, \bar{\mathsf{a}}'^{\downarrow}_{k'}), \ldots, \sigma^2_n(\bar{\mathsf{a}}'^{\downarrow}_1, \ldots, \bar{\mathsf{a}}'^{\downarrow}_n)\big)$$

in analogy to Case (A2).

We next consider the four possible combinations of the above cases.

Suppose Cases (A1) and (B1) apply. We then have $\mathcal{A}, \beta^{\downarrow} \models A$ and $\mathcal{A}, \beta'^{\downarrow} \not\models B$. But by Equation (4.3) we also have

$$\beta^{\downarrow}(A) = P\big(\beta^{\downarrow}(s_1), \ldots, \beta^{\downarrow}(s_m)\big) = P\big(\beta'^{\downarrow}(t_1), \ldots, \beta'^{\downarrow}(t_m)\big) = \beta'^{\downarrow}(B).$$

In other words, the structure $\mathcal{A}$ is inconsistent, which contradicts our assumptions.

Next, suppose that Cases (A2) and (B1) apply. Since B belongs to At_0, we have $t_1, \ldots, t_m \in \bar{\mathrm{y}}$ and, hence, $\beta'(t_\ell) \in \mathsf{D}_{-1}$ for every ℓ. This means that A cannot contain any variables $u \in \bar{\mathrm{u}}$, for otherwise there would be some ℓ with $s_\ell = u$ leading to $\beta(s_\ell) \in \mathsf{D}_i$ with $i \geq 0$ and $\beta'(t_\ell) \in \mathsf{D}_{-1}$, which would contradict our assumption $\beta(s_\ell) = \beta'(t_\ell)$. This means, we have

$$\mathcal{A} \models A\big((\tau^1_1(\bar{\mathsf{a}}_1))^{\downarrow}, \ldots, (\tau^1_{k-1}(\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{k-1}))^{\downarrow}, \bar{\mathsf{a}}^{\downarrow}_k\big),$$

which translates to $\mathcal{A}, \beta^{\downarrow} \models A(\bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_{k-1}, \bar{\mathrm{x}}_k)$. As in the previous case, $\mathcal{A}, \beta^{\downarrow} \models A$ together with $\mathcal{A}, \beta'^{\downarrow} \not\models B$ leads to a contradiction.

The combination of Case (A1) and Case (B2) also leads to a contradiction.

Finally, suppose that Case (A2) applies together with Case (B2).

First, we make two more observations:

(I) Consider any two variables $y_1 \in \bar{\mathrm{y}}_{k_1}, y_2 \in \bar{\mathrm{y}}_{k_2}$. By definition of τ, $\beta(y_1) = \beta'(y_2)$ entails
   (a) $\big(\beta(y_1)\big)^{\downarrow} = \big(\beta'(y_2)\big)^{\downarrow}$,
   (b) $k_1 = k_2$ and $y_1 = y_2$,
   (c) $\langle \bar{\mathsf{a}}^{\downarrow}_1, \ldots, \bar{\mathsf{a}}^{\downarrow}_{k_1} \rangle \sim \langle \bar{\mathsf{a}}'^{\downarrow}_1, \ldots, \bar{\mathsf{a}}'^{\downarrow}_{k_1} \rangle$, and
   (d) $T_i = T'_i$ and $\bar{\mathsf{c}}_i = \bar{\mathsf{c}}'_i$ for every i with $1 \leq i \leq k_1$.

   Moreover, the definition of the τ^1_k also entails that $\big(\beta(y_1)\big)^{\downarrow} = \gamma(y_1)$ and $\big(\beta'(y_2)\big)^{\downarrow} = \gamma'(y_2)$.

(II) Consider any two variables $u_1 \in \bar{\mathrm{u}}_{k_1}, u_2 \in \bar{\mathrm{u}}_{k_2}$. By definition of τ, $\beta(u_1) = \beta'(u_2)$ entails
   (a) $\big(\beta(u_1)\big)^{\downarrow} = \big(\beta'(u_2)\big)^{\downarrow}$,
   (b) $k_1 = k_2$ and $u_1 = u_2$ and $x = x'$ and $u_1, u_2 \in U_x$,
   (c) $k = k' \leq k_1$,
   (d) $\langle \bar{\mathsf{a}}^{\downarrow}_1, \ldots, \bar{\mathsf{a}}^{\downarrow}_{k_1} \rangle \sim \langle \bar{\mathsf{a}}'^{\downarrow}_1, \ldots, \bar{\mathsf{a}}'^{\downarrow}_{k_1} \rangle$,
   (e) $T_i = T'_i$ and $\bar{\mathsf{c}}_i = \bar{\mathsf{c}}'_i$ for every i with $1 \leq i \leq k_1$,
   (f) $\beta(x) \in \mathcal{T}_{\sigma^1}$ if and only if $\beta'(x) \in \mathcal{T}_{\sigma^1}$, and
   (g) if $\beta(x), \beta'(x) \in \mathcal{T}_{\sigma^1}$ then $\beta(x) = \beta'(x)$ and $\beta(x) = \gamma(x)$ and $\beta'(x) = \gamma'(x)$.

   Moreover, the definition of τ also entails that $\big(\beta(u_1)\big)^{\downarrow} = \gamma(u_1)$ and $\big(\beta'(u_2)\big)^{\downarrow} = \gamma'(u_2)$.

We now derive contradictions the following cases (the remaining cases can be treated analogously).

Suppose there is some ℓ such that $s_\ell = u \in \bar{\mathrm{u}}_{k_1}$ and $t_\ell = u' \in \bar{\mathrm{u}}_{k_2}$. According to Observation (II), this entails, among other things, that $x = x'$.

Now, let k_* be the largest index for which there is such an ℓ with $s_\ell = u \in \bar{\mathrm{u}}_{k_*}$ and $t_\ell = u' \in \bar{\mathrm{u}}_{k_*}$. Then, we know that $k_* \geq k$. Moreover, $T_i = T'_i$ and $\bar{\mathsf{c}}_i = \bar{\mathsf{c}}'_i$ for every i with $1 \leq i \leq k_*$. Hence, γ and γ' coincide on all variables in $\bar{\mathrm{y}}_i$, $\bar{\mathrm{x}}_i$, $\bar{\mathrm{u}}_i$ with $1 \leq i \leq k_*$.

By definition of τ, there is some D_i with $i \geq 0$ to which all elements $\beta(u'')$ with $u'' \in \mathrm{vars}(A) \cap U_x$ and all $\beta'(u'')$ with $u'' \in \mathrm{vars}(B) \cap U_x$ belong. Moreover, $\beta(x)$ and $\beta'(x)$ cannot belong to this D_i. Therefore, there is no ℓ such that s_ℓ is a variable from $\bar{\mathrm{u}}$ and t_ℓ is not from $\bar{\mathrm{u}}$ or vice versa. Hence, every $u \in \bar{\mathrm{u}}$ that occurs in A or in B stems from $\bar{\mathrm{u}}_1 \cup \ldots \cup \bar{\mathrm{u}}_{k_*}$.

Consequently, γ and γ' indeed coincide on all variables that occur in A or in B. This leads to $\mathcal{A}, \gamma \not\models B$ (as $\mathcal{A}, \gamma' \not\models B$ and $\gamma(B) = \gamma'(B)$) and $\mathcal{A}, \gamma' \models A$ (as $\mathcal{A}, \gamma \models A$ and $\gamma'(A) = \gamma(A)$). Moreover, by definition of τ, β and β' coincide on all variables in $\bar{\mathrm{y}}_i, \bar{\mathrm{u}}_i$ with $1 \leq i \leq k_*$.

Due to the above observations and because of definition of τ, there are two variables $y_1, y_2 \in \bar{\mathrm{y}}$ such that for all arguments s_ℓ and t_ℓ one of the following cases applies:

$s_\ell = y = t_\ell$ for a certain variable $y \in \bar{\mathrm{y}}$,

$s_\ell = x$ and $t_\ell = y_2$,

$s_\ell = y_1$ and $t_\ell = x$,

$s_\ell = x = t_\ell$, or

$s_\ell = u = t_\ell$ for a certain variable $u \in \bar{\mathrm{u}}$.

Let L be the set of all indices ℓ such that $s_\ell = x$ and $t_\ell = y_2 \in \bar{\mathrm{y}}$. Similarly, let L' be the set of all indices ℓ such that $s_\ell = y_1 \in \bar{\mathrm{y}}$ and $t_\ell = x$. Suppose L is nonempty, i.e. there is some ℓ with $s_\ell = x$ and $t_\ell = y_2$. Then, our assumption $\beta(x) = \beta(s_\ell) = \beta'(t_\ell) = \beta'(y_2)$ entails $\beta(x) \in \mathcal{T}_{\sigma^1}$. This leads to the following chain of equations: $\gamma(x) = \beta^{\downarrow}(x) = \beta'^{\downarrow}(y_2) = \gamma'(y_2)$. Hence, for every $\ell \in L$ we have $\gamma(s_\ell) = \gamma(x) = \gamma'(y_2) = \gamma(y_2) = \gamma(t_\ell)$. Symmetrically, we have $\gamma'(x) = \beta'^{\downarrow}(x) = \beta^{\downarrow}(y_1) = \gamma(y_1)$ and $\gamma(s_\ell) = \gamma(y_1) = \gamma'(x) = \gamma(x) = \gamma(t_\ell)$ for every $\ell \in L'$, if L' in nonempty.

Put together, the above observations entail

$$\gamma(A) = P\big(\gamma(s_1), \ldots, \gamma(s_m)\big) = P\big(\gamma(t_1), \ldots, \gamma(t_m)\big) = \gamma(B).$$

Recall that we have already inferred $\mathcal{A}, \gamma \models A$ and also $\mathcal{A}, \gamma \not\models B$. Together with the equality of A and B under γ, this yields a contradiction with our assumption that $\mathcal{A}$ is a well defined structure.

Suppose we have $t_\ell \notin \bar{\mathrm{u}}$ for every ℓ for which $s_\ell \in \bar{\mathrm{u}}$. Further suppose, there is indeed some $u \in \bar{\mathrm{u}}$ such that $s_\ell = u$. Since $\beta(s_\ell) = \beta'(t_\ell)$, the definition of τ entails that $t_\ell = x'$. Let D_i be the subdomain that contains $\beta'(x')$ and thus also $\beta(u)$. Then, $\beta(x) \in \mathsf{D}_{(i-1 \bmod 3)}$ and for any $u' \in \mathrm{vars}(B) \cap \bar{\mathrm{u}}$ we have $\beta'(u') \in \mathsf{D}_{(i+1 \bmod 3)}$, which entails $\beta'(u') \neq \beta(x)$. Hence, B cannot contain any variable from $\bar{\mathrm{u}}$. Moreover, u is the only variable from $\bar{\mathrm{u}}$ that occurs in A, since for any $u'' \in \bar{\mathrm{u}} \setminus \{u\}$ with $u'' = s_{\ell'}$ we would have $\beta(u'') \neq \beta(u)$, on the one hand, but $\beta(u'') = \beta(s_{\ell'}) = \beta'(t_{\ell'}) = \beta'(x') = \beta(u)$ on the other hand.

Therefore and by definition of τ, there is some $y_2 \in \bar{\mathrm{y}}$ such that for all arguments s_ℓ, t_ℓ one of the following cases applies:

$s_\ell = y = t_\ell$ for a certain variable $y \in \bar{\mathrm{y}}$,

$s_\ell = x$ and $t_\ell = y_2$, or

$s_\ell = u$ and $t_\ell = x'$.

Let k_* be the largest index for which there is some ℓ with $s_\ell = t_\ell = y \in \bar{\mathrm{y}}_{k_*}$ (let $k_* := 0$ if no such index exists). By virtue of Observation (I), we then get $T_i = T'_i$ and $\bar{\mathsf{c}}_i = \bar{\mathsf{c}}'_i$ for every i with $1 \leq i \leq k_*$. Hence, γ and γ' coincide on all variables in $\bar{\mathrm{y}}_i$ with $1 \leq i \leq k_*$. By definition of β', this also applies to γ and $\beta'^{\downarrow}$.

If there is some ℓ such that $s_\ell = x$ and $t_\ell = y_2$, then we have $\beta^{\downarrow}(x) = \beta^{\downarrow}(s_\ell) = \beta'^{\downarrow}(t_\ell) = \beta'^{\downarrow}(y_2) \in \mathcal{T}_{\sigma^1}$. This leads to $\beta^{\downarrow}(x) = \gamma(x)$. Therefore, we have $\gamma(x) = \beta'^{\downarrow}(y_2)$.

Finally, consider the ℓ for which $s_\ell = u$ and $t_\ell = x'$. Since $\gamma(u) = \beta^{\downarrow}(u) = \beta'^{\downarrow}(x')$, we have $\gamma(u) = \beta'^{\downarrow}(x')$.

Consequently, we have all pieces together to conclude
$$P\big(\gamma(s_1),\ldots,\gamma(s_m)\big) = P\big(\beta'^{\downarrow}(s_1),\ldots,\beta'^{\downarrow}(s_m)\big).$$
In other words, A under γ equals B under $\beta'^{\downarrow}$. Since we have $\mathcal{A},\gamma \models A$ and $\mathcal{A},\beta'^{\downarrow} \not\models B$ (cf. Case (B2) together with the observation that B does not contain any variable from $\bar{\mathrm{u}}$), this leads to a contradiction with the assumption that the structure $\mathcal{A}$ is well defined.

Suppose neither A nor B contain any variable from $\bar{\mathrm{u}}$. According to the statements derived for Case (A2) and Case (B2), we have
$$\mathcal{A} \models A\big((\tau_1^1(\bar{\mathsf{a}}_1))^{\downarrow},\ldots,(\tau_{k-1}^1(\bar{\mathsf{a}}_1,\ldots,\bar{\mathsf{a}}_{k-1}))^{\downarrow},\bar{\mathsf{a}}_k^{\downarrow}\big).$$
and
$$\mathcal{A} \not\models B\big((\tau_1^1(\bar{\mathsf{a}}'_1))^{\downarrow},\ldots,(\tau_{k'-1}^1(\bar{\mathsf{a}}'_1,\ldots,\bar{\mathsf{a}}'_{k'-1}))^{\downarrow},\bar{\mathsf{a}}'^{\downarrow}_{k'}\big).$$
Put differently, $\mathcal{A},\beta^{\downarrow} \models A$ and $\mathcal{A},\beta'^{\downarrow} \not\models B$. As we, by Equation (4.3), already know that A under $\beta^{\downarrow}$ equals B under $\beta'^{\downarrow}$, this contradicts our assumption that $\mathcal{A}$ is a well-defined structure. ◊

Claim IV: Under $\mathcal{B}$ the strategy τ is satisfying for φ.

Proof: Let $\bar{\mathsf{a}}_1,\ldots,\bar{\mathsf{a}}_n$ be any sequence of tuples with $\bar{\mathsf{a}}_i \in \mathsf{B}^{|\bar{\mathrm{x}}_i|}$ for every i. Let β be the variable assignment that maps every $\bar{\mathrm{x}}_i$ to $\bar{\mathsf{a}}_i$, every $\bar{\mathrm{y}}_i$ to $\tau_i^1(\bar{\mathsf{a}}_1,\ldots,\bar{\mathsf{a}}_i)$, and every $\bar{\mathrm{u}}_i$ to $\tau_i^2(\bar{\mathsf{a}}_1,\ldots,\bar{\mathsf{a}}_i)$. We intend to show $\mathcal{B},\beta \models \psi$.

Let $\bar{\mathsf{c}}_1,\ldots,\bar{\mathsf{c}}_n$ be the distinguished representative of the equivalence class $[\langle\bar{\mathsf{a}}_1^{\downarrow},\ldots,\bar{\mathsf{a}}_n^{\downarrow}\rangle]_{\sim}$. Let γ be the variable assignment that maps every $\bar{\mathrm{x}}_i$ to $\bar{\mathsf{c}}_i$, every $\bar{\mathrm{y}}_i$ to $\sigma_i^1(\bar{\mathsf{c}}_1,\ldots,\bar{\mathsf{c}}_i)$, and every $\bar{\mathrm{u}}_i$ to $\sigma_i^2(\bar{\mathsf{c}}_1,\ldots,\bar{\mathsf{c}}_i)$. Since we assume σ to be a satisfying strategy for φ under $\mathcal{A}$, we have $\mathcal{A},\gamma \models \psi$. Consider any atom A that occurs in φ. By definition of $\mathcal{B}$, we have $\mathcal{B},\beta \models A$ if and only if $\mathcal{A},\gamma \models A$. This together with $\mathcal{A},\gamma \models \psi$ entails $\mathcal{B},\beta \models \psi$.

Consequently, under $\mathcal{B}$ τ is satisfying for φ. ◊

$\mathcal{T}_\tau$, $\mathcal{T}_k$ Let $\mathcal{T}_\tau \subseteq \mathsf{B}$ be the *target set of* τ, given by $\mathcal{T}_\tau := \bigcup_{k=1}^n \mathcal{T}_k$ where
$$\mathcal{T}_k := \big\{\mathsf{b}\in\mathsf{B} \;\big|\; \text{there are tuples } \bar{\mathsf{a}}_1,\ldots,\bar{\mathsf{a}}_k \text{ such that } \tau_k^1(\bar{\mathsf{a}}_1,\ldots,\bar{\mathsf{a}}_k) = \langle\ldots,\mathsf{b},\ldots\rangle \text{ or } \tau_k^2(\bar{\mathsf{a}}_1,\ldots,\bar{\mathsf{a}}_k) = \langle\ldots,\mathsf{b},\ldots\rangle \;\big\}.$$
Finiteness of the set $\mathcal{T}_\tau$ follows from the following observations: Let $\mathcal{S}_\sigma$ be the set $\mathcal{S}_\sigma := \bigcup_{k=1}^n \mathcal{S}_k$ where
$$\mathcal{S}_k := \big\{\mathsf{b}\in\mathsf{A} \;\big|\; \text{there is some equivalence class } C \in (\underline{\mathsf{A}}_1\cup\ldots\cup\underline{\mathsf{A}}_n)/_\sim \text{ and its distinguished representative } \langle\bar{\mathsf{c}}_1,\ldots,\bar{\mathsf{c}}_k\rangle := \alpha_C \text{ such that } \sigma_k^1(\bar{\mathsf{c}}_1,\ldots,\bar{\mathsf{c}}_k) = \langle\ldots,\mathsf{b},\ldots\rangle \text{ or } \sigma_k^2(\bar{\mathsf{c}}_1,\ldots,\bar{\mathsf{c}}_k) = \langle\ldots,\mathsf{b},\ldots\rangle \;\big\}.$$
Since the equivalence relation $\sim$ induces only finitely many equivalence classes C on the set $\underline{\mathsf{A}}_1\cup\ldots\cup\underline{\mathsf{A}}_n$, the set $\mathcal{S}_\sigma$ is finite. By definition of the mappings τ_k^1 and τ_k^2, we have $\mathsf{b}^{\downarrow}\in\mathcal{S}_\sigma$ for every $\mathsf{b}\in\mathcal{T}_\tau$. The domain of $\mathcal{B}$ consists of finitely many subdomains D_i, each of which comprises finitely many copies of the original domain A. Hence, there are only finitely many domain elements $\mathsf{b}\in\mathsf{B}$ with $\mathsf{b}^{\downarrow}\in\mathcal{S}_\sigma$. Altogether, this entails that the target set $\mathcal{T}_\tau$ is finite.

Notice that $\mathcal{T}_\tau$ coincides with the domain of $\mathcal{B}|_\tau$. Hence, by Lemma 4.2.3, $\mathcal{B}|_\tau$ is a finite model of φ and it can thus serve as the sought model $\mathcal{C}$. Finally, we give an upper bound on the size of $\mathcal{C}$'s domain. First, we bound the number of distinct sequences of fingerprints $\langle T_1,\ldots,T_k\rangle \in \mathrm{im}_\sigma(\underline{\lambda}_{1,0})\times\ldots\times\mathrm{im}_\sigma(\underline{\lambda}_{k,0})$. Let $\mathrm{At}_{\bar{\mathrm{x}}_i}$ denote the set $\bigcup_{x\in\bar{\mathrm{x}}_i}\mathrm{At}_x$. For every i with $1\le i\le k$ we have

$$\big|\mathrm{im}_\sigma(\underline{\lambda}_{i,0})\big| = \prod_{x\in\bar{\mathrm{x}}_i}\big|\mathrm{im}_\sigma(\lambda_{x,0})\big| \le \prod_{x\in\bar{\mathrm{x}}_i} 2^{\uparrow n-i+2}(|\mathrm{At}_x|) \le 2^{\uparrow n-i+2}\Big(\sum_{x\in\bar{\mathrm{x}}_i}|\mathrm{At}_x|\Big) \le 2^{\uparrow n-i+2}(|\mathrm{At}_{\bar{\mathrm{x}}_i}|)\ .$$

This leads to the following bound on the number of sequences of fingerprints:

$$\begin{aligned}\sum_{k=1}^{n}\big|\mathrm{im}_\sigma\big(\underline{\lambda}_{1,0}\big)\times\ldots\times\mathrm{im}_\sigma\big(\underline{\lambda}_{k,0}\big)\big| &\leq \sum_{k=1}^{n}\prod_{i=1}^{k}2^{\uparrow n-i+2}(|\mathrm{At}_{\bar{\mathsf{x}}_i}|)\\ &\leq \sum_{k=1}^{n}2^{\uparrow n+1}\Big(\sum_{i=1}^{k}|\mathrm{At}_{\bar{\mathsf{x}}_i}|\Big)\\ &\leq n\cdot 2^{\uparrow n+1}(|\mathrm{At}|)\ .\end{aligned}$$

Since σ is λ-semi-uniform, this entails an upper bound regarding the cardinality of the set $\mathcal{T}_{\sigma^1}$: $|\mathcal{T}_{\sigma^1}|\leq|\bar{\mathsf{y}}|\cdot n\cdot 2^{\uparrow n+1}(|\mathrm{At}|)$. The number of equivalence classes induced by the relation $\sim$ depends on the cardinality of $\mathcal{T}_{\sigma^1}$ and the number of sequences of fingerprints:

$$\begin{aligned}\big|(\underline{\mathsf{A}}_1\cup\ldots\cup\underline{\mathsf{A}}_n)/_\sim\big| &\leq \Big(\sum_{k=1}^{n}(|\mathcal{T}_{\sigma^1}|+1)^{|\bar{\mathsf{x}}_1|+\ldots+|\bar{\mathsf{x}}_k|}\Big)\cdot n\cdot 2^{\uparrow n+1}(|\mathrm{At}|)\\ &\leq n^2\cdot(|\mathcal{T}_{\sigma^1}|+1)^{|\bar{\mathsf{x}}|}\cdot 2^{\uparrow n+1}(|\mathrm{At}|)\ .\end{aligned}$$

Given this, we can bound the number of elements in the set $\mathcal{S}_\sigma$:

$$|\mathcal{S}_\sigma|\leq|\mathcal{T}_{\sigma^1}|+\big|(\underline{\mathsf{A}}_1\cup\ldots\cup\underline{\mathsf{A}}_n)/_\sim\big|\cdot|\bar{\mathsf{u}}|.$$

The last component that we need to establish an upper bound on the cardinality of the set $\mathcal{T}_\tau$, is the number of copies of A that are used as building blocks for the sets $\mathsf{D}_{-1},\mathsf{D}_0,\mathsf{D}_1,\mathsf{D}_2$. An upper bound for this number is $|\bar{\mathsf{y}}|\cdot\big|(\underline{\mathsf{A}}_1\cup\ldots\cup\underline{\mathsf{A}}_n)/_\sim\big|+3\cdot|\bar{\mathsf{u}}|\cdot\big|(\underline{\mathsf{A}}_1\cup\ldots\cup\underline{\mathsf{A}}_n)/_\sim\big|$. All in all we obtain the following upper bound:

$$|\mathcal{T}_\tau|\leq|\mathcal{S}_\sigma|\cdot(|\bar{\mathsf{y}}|+3\cdot|\bar{\mathsf{u}}|)\cdot\big|(\underline{\mathsf{A}}_1\cup\ldots\cup\underline{\mathsf{A}}_n)/_\sim\big|\leq\big(p(\mathrm{len}(\varphi))\big)^{2|\bar{\mathsf{x}}|}\cdot\big(2^{\uparrow n+1}(|\mathrm{At}|)\big)^{2|\bar{\mathsf{x}}|+2},$$

where p is some polynomial and where we assume $|\bar{\mathsf{x}}|\geq 1$ and $|\mathcal{T}_{\sigma^1}|\geq 1$. □

In analogy to Theorem 4.2.11, the bound given in Lemma 4.3.6 and Theorem 4.3.5 could be refined using a notion of *degree* for GAF sentences, cf. Definition 4.2.9. For φ the *degree* $\partial_{\mathrm{GAF}}(\varphi)$ is the smallest nonnegative integer m such that for every At_x, $x\in\bar{\mathsf{x}}_i$, there are at most m distinct indices $j_1,\ldots,j_m$ with $i\leq j_1<\ldots<j_m\leq n$ such that $\bar{\mathsf{u}}_{j_\ell}\cap\mathrm{vars}(\mathrm{At}_x)\neq\emptyset$. $\partial_{\mathrm{GAF}}(\varphi)$

We conclude this section with an example that illustrates the construction of a finite model from the proof of Theorem 4.3.5.

Example 4.3.7. *Consider again the sentence φ and its model $\mathcal{A}$ from Example 4.3.1:*

$$\varphi:=\exists z\forall x\exists y_1y_2.\,Q(z)\wedge\neg R(x,x)\wedge R(x,y_1)\wedge Q(y_1)\wedge R(x,y_2)\wedge\neg Q(y_2)\ .$$

We observe that all elements in A have the same fingerprint. More precisely, for every $\mathsf{a}\in\mathsf{A}$ we have $\lambda_{x,0}\big(\sigma_1^1(),\mathsf{a}\big)=\mathcal{P}\big(\mathrm{At}_x\setminus\{R(x,x)\}\big)$ and $\lambda_{x,1}\big(\sigma_1^1(),\mathsf{a},\sigma_2^2(\mathsf{a})\big)=\{R(x,y_1),Q(y_1),R(x,y_2)\}$. As we have already pointed out, none of the finite substructures of $\mathcal{A}$ is a model of φ. Nevertheless, we can use parts of $\mathcal{A}$ as a blueprint for constructing the following structure $\mathcal{C}$ that is finite and satisfies φ. The domain of $\mathcal{C}$ consists of the disjoint union $\mathsf{D}'_{-1}\uplus\mathsf{D}'_0\uplus\mathsf{D}'_1\uplus\mathsf{D}'_2=\{0\}\uplus\{1,2,3,4\}\uplus\{3,4\}\uplus\{3,4\}$, which results in $\mathsf{C}:=\{0,1,2,3,4,3',4',3'',4''\}$. The interpretation of the predicate symbols Q and R under $\mathcal{C}$ is depicted in Figure 4.2. The canonical satisfying strategy τ for φ under $\mathcal{C}$ is indicated by the arrows and their annotation.

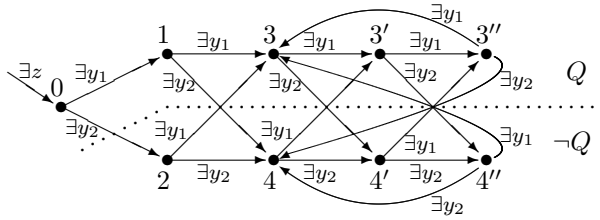


Figure 4.2: Illustration of the structure $\mathcal{C}$. The arrows indicate $R^{\mathcal{C}}$, as in Figure 4.1. The annotated existential quantifiers indicate the elements selected by the strategy τ. All elements above the dotted line belong to $Q^{\mathcal{C}}$, the others do not.

Chapter 5

Computational Complexity of SF-Sat and GBSR-Sat

The investigation of the classical decision problem has not stopped at the point where fragments had been found to be decidable. The computational complexity of the satisfiability problem associated with a decidable class is of interest as well. Systematic investigations in this direction were started in the late 1970s [Lew78, Lew80, Für81, Pla84, DL84a, DL84b].[1] There is an easy route to upper bounds whenever we know that the fragment under consideration enjoys a small model property. The following lemma links bounds regarding the size of models with the computing time that is required to decide satisfiability.

Proposition 5.0.1 ([Lew80], Proposition 3.2, see also [BGG97], Proposition 6.0.4)**.** *Let φ be a first-order sentence in prenex normal form containing n universally quantified variables. The question whether φ has a model of cardinality m can be decided nondeterministically in time $p\big(m^n \cdot \mathrm{len}(\varphi)\big)$ for some polynomial p.*

With this lemma at hand, it is enough to prove a small model property for a given class of first-order sentences, in order to bound the worst-case time complexity of the corresponding satisfiability problem from above. This approach yields good upper bounds in cases where the Boolean structure of the considered sentences is not restricted. As we shall see later, namely in Sections 5.1 and 5.2, certain prefix classes of Horn or Krom sentences are complete for deterministic time complexity classes or for space complexity classes. Then, a small model property alone does not immediately lead to good upper bounds and one needs more sophisticated arguments.

In the present chapter, we aim to investigate the computational complexity of SF-Sat and GBSR-Sat. Considerations regarding the other decidable fragments that we have introduced in Chapter 3 remain future work.[2] Recall that we have shown several small model properties for SF and GBSR, cf. Theorem 3.2.6 for SF and Corollary 3.5.4 and Theorem 4.2.11 for GBSR. More concretely, any satisfiable SF or GBSR sentence φ with degree k has a model whose domain is of a size that is at most k-fold exponential in the length of φ. Therefore, Proposition 5.0.1 immediately supplies us with upper bounds regarding the computational complexity of SF-Sat and GBSR-Sat. If we use the (various) degrees of interaction of variables as parameters, we obtain upper bounds for infinitely many subclasses $\mathrm{SF}_{\leq k}$ and $\mathrm{GBSR}_{\leq k}$ with $k \geq 0$, as depicted in Figure 5.1. The set $\mathrm{SF}_{\leq k}$ collects all SF formulas φ with $\partial_\exists(\varphi) \leq k$ or $\partial_\forall(\varphi) \leq k$ or both. The satisfiability problem associated with any fragment $\mathrm{SF}_{\leq k}$ is denoted by $\mathrm{SF}_{\leq k}$-Sat. The set $\mathrm{GBSR}_{\leq k}$ is the

$\mathrm{SF}_{\leq k}$, $\mathrm{GBSR}_{\leq k}$

[1] The foundations of computational complexity theory can be found in many standard textbooks, e.g. [HU79, Pap94, Gol08, AB09]

[2] It seems safe to assume that the satisfiability problem for each of the decidable relational first-order fragments introduced in Chapter 3 lies in Tower (cf. Definition 5.0.2). Intuitively, the translations of the separated versions into their respective base fragment lead to a blowup that is at most $p(n)$-fold exponential in the length n of the formula for some polynomial p. Moreover, the satisfiability problems of the base fragments usually do not go beyond 2-ExpTime, except for FL, whose satisfiability problem lies in Tower. But even SFL-Sat should lie in Tower.

collection of all GBSR sentences φ with $\partial_{\exists\forall}(\varphi) \leq k$ or $\partial_{\forall}(\varphi) \leq k-1$ or both. $\mathrm{GBSR}_{\leq k}$-Sat denotes the satisfiability problem associated with $\mathrm{GBSR}_{\leq k}$. Our results then entail for every positive k that $\mathrm{SF}_{\leq k}$-Sat and $\mathrm{GBSR}_{\leq k}$-Sat are in k-NExpTime. The decision problems $\mathrm{SF}_{\leq 0}$-Sat and $\mathrm{GBSR}_{\leq 0}$-Sat belong to NExpTime. If we consider SF or GBSR without universal quantifiers, we obtains the fragments $\exists^*$-SF and $\exists^*$-GBSR, which coincide with the *existential fragment of relational first-order logic*, see also Section 5.1. The corresponding decision problems $\exists^*$-SF-Sat and $\exists^*$-GBSR-Sat coincide as well and belong to NP.

$\exists^*$-SF, $\exists^*$-GBSR

We shall complement these upper bounds with matching lower bounds, which in the end will lead to a hierarchy of computationally hard satisfiability problems that are even complete for k-NExpTime. Recall that $\textsc{Elementary} := \bigcup_{k\geq 1} k\text{-ExpTime}$ is the complexity class containing k-ExpTime, k-NExpTime, and k-ExpSpace for every positive k. The unrestricted problems SF-Sat and GBSR-Sat lie even beyond Elementary. Indeed, both are complete for the complexity class Tower, which contains Elementary but is slightly larger [Sch16]. In contrast to Elementary, Tower contains problems that are complete for the class.

Elementary

Definition 5.0.2 (Tower, [Sch16])**.** *First, we define the following functions $F_i : \mathbb{N} \to \mathbb{N}$ for $i = 0, 1, 2, 3$, where $F_j^k(n) := \underbrace{F_j\big(F_j\big(\ldots F_j(n)\ldots\big)\big)}_{k \text{ times}}$ denotes k-fold application of F_j to the argument n:*

$$
\begin{aligned}
F_0(n) &:= n + 1 \ , \\
F_1(n) &:= F_0^{n+1}(n) = 2n + 1 \ , \\
F_2(n) &:= F_1^{n+1}(n) = 2^{n+1} \cdot (n+1) - 1 \ , \\
F_3(n) &:= F_2^{n+1}(n) \ > \ 2^{\uparrow n}(2) \ .
\end{aligned}
$$

Let FElem *denote the set of elementary functions $f : \mathbb{N} \to \mathbb{N}$, i.e. of all functions that can be bounded from above by $2^{\uparrow k}(n)$ for some constant k. The complexity class* Tower *is defined by*

$$
\textsc{Tower} := \bigcup_{f(n)\in \mathrm{FElem}} \mathrm{DTime}\big(F_3(f(n))\big) \ .
$$

Since F_3 is a non-elementary function that grows faster than $2^{\uparrow n}(2)$ ([Sch16], page 4), the complexity class Tower properly contains Elementary.

We now state the main result of the present chapter.

Theorem 5.0.3. *The decision problems $\exists^*$-SF-Sat and $\exists^*$-GBSR-Sat are NP-complete and the problems $\mathrm{SF}_{\leq 0}$-Sat and $\mathrm{GBSR}_{\leq 0}$-Sat are NExpTime-complete. For any positive integer k the problems $\mathrm{SF}_{\leq k}$ and $\mathrm{GBSR}_{\leq k}$ are complete for k-fold nondeterministic exponential time. The problems SF-Sat and GBSR-Sat are Tower-complete (with respect to elementary reductions).*

Proof. The upper bounds and hardness results for $\exists^*$-SF-Sat and $\exists^*$-GBSR-Sat will be derived in Section 5.1, in particular in Proposition 5.1.4. The upper bounds for $\mathrm{SF}_{\leq k}$-Sat and $\mathrm{GBSR}_{\leq k}$-Sat with nonnegative k follow from Proposition 5.0.1 together with the small model properties given in Theorem 3.2.6, Corollary 3.5.4, and Theorem 4.2.11. These bounds also lead to the observation that the unrestricted problems SF-Sat and GBSR-Sat belong to Tower.

In Section 5.3 we derive corresponding lower bounds. More precisely, Theorem 5.3.11 establishes k-NExpTime-hardness for $\mathrm{SF}_{\leq k}$-Sat for every positive k and thus also for SF-Sat. According to [Sch16], Section 3.1, k-NExpTime-hardness of SF-Sat for every positive k entails hardness for the class Tower. NExpTime-hardness for $\mathrm{SF}_{\leq 0}$ follows from NExpTime-hardness of BSR-Sat. As for every nonnegative k the set $\mathrm{SF}_{\leq k}$-Sat is a subproblem of $\mathrm{GBSR}_{\leq k}$-Sat, we get the same lower bounds for every $\mathrm{GBSR}_{\leq k}$-Sat and for GBSR-Sat, respectively. □

Recall that we have already derived a general reducibility result in Theorem 3.3.11, which entails that SF-Sat inherits computational hardness from the satisfiability problems of other first-order fragments that exhibit a small model property with a bound $2^{\uparrow c\cdot \mathrm{len}(\varphi)}(d \cdot \mathrm{len}(\varphi))$. Although this already entails that SF-Sat is, e.g., as hard as the satisfiability problem associated with the fluted

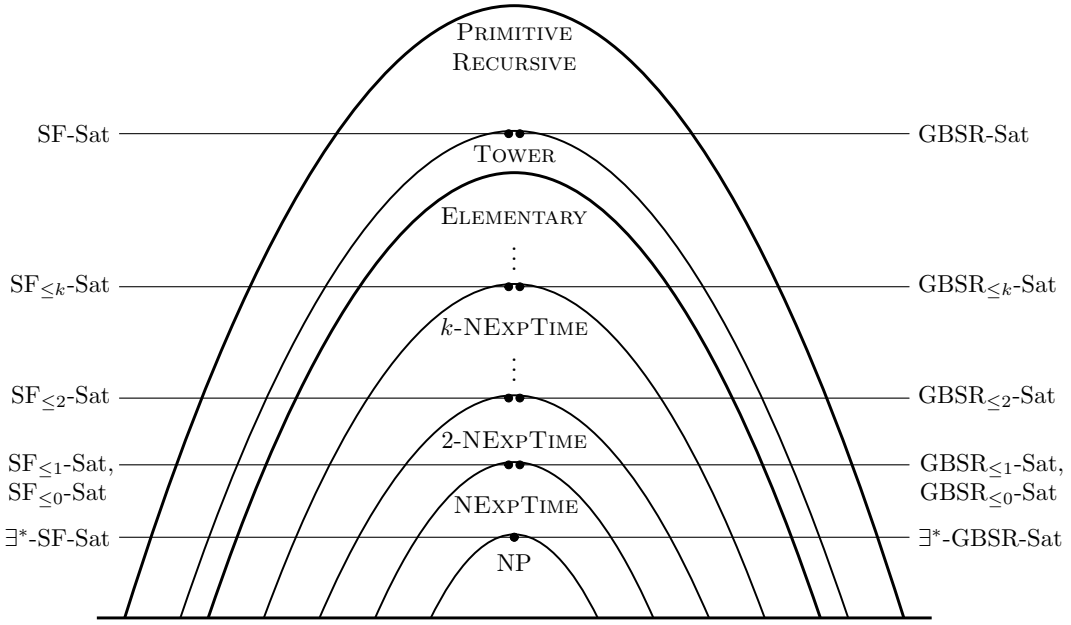


Figure 5.1: The computational complexity of the subfragments of SF and GBSR scale over the major nondeterministic complexity classes in Elementary, while the unrestricted problems SF-Sat and GBSR-Sat even go beyond.

fragment, which is known to be non-elementary [PST16], Theorem 5.0.3 is much more accurate and yields better lower bounds than what we could derive from Theorem 3.3.11. Moreover, the lower-bound proof for SF-Sat that will be presented in Section 5.3 is more instructive in that it provides a better understanding of how large models can be enforced with SF sentences.

It is worth mentioning that Theorem 5.0.3 adequately accounts for the known complexity of MFO-Sat. This works in spite of the fact that monadic sentences may contain arbitrarily nested alternating quantifiers. For every MFO sentence φ we trivially have $\partial_{\exists}(\varphi) = 1$, since all occurring predicate symbols have an arity of at most one. Theorem 5.0.3 entails that MFO-Sat is in NExpTime, which is well known. Moreover, we have introduced the strongly separated fragment (SSF) as the set of SF sentences that have degree one, see Definition 3.2.3 and the short paragraph preceding it. SSF contains BSR and MFO as subfragments and, by Theorem 5.0.3, shares their computational complexity.

Corollary 5.0.4. *The satisfiability problem for SSF is* NExpTime*-complete.*

Theorem 5.0.3 establishes two hierarchies of computationally hard problems that are complete for infinitely many subclasses of Elementary and as a whole form a problem that is complete for Tower. The overall structure is depicted in Figure 5.1.

Apparently, non-elementary satisfiability problems are not very widespread among the decidable fragments of classical relational fist-order logic known today. We have argued that SF and GBSR fall into this category. To the present author's knowledge, the only known companion in this respect is the fluted fragment. Indeed, Pratt-Hartmann, Szwast, and Tendera showed in [PST16] that satisfiability of fluted sentences with at most $2k$ variables is k-NExpTime-hard. Moreover, they argue that satisfiability of fluted sentences with at most k variables lies in k-NExpTime. Although a significant gap between these lower and upper bounds remains to be closed, the fluted fragment seems to comprise a similar hierarchy of hard problems as SF does, and the unrestricted problem FL-Sat is Tower-complete as well. Another candidate for a first-order fragment with

such a high computational complexity might be Maslov's fragment K.[3]

Remark 5.0.5. *In computational complexity theory, several hierarchies have been defined where the hardness of problems is suspected to grow with the number of quantifier alternations that are admitted in the formal problem description. One example is the* polynomial-time hierarchy (PH) *(cf. Definition 3.8 in [Gol08], or Definition 5.3 in [AB09]). Another example is the* exponential-time hierarchy *(see, e.g. [Har87]).*[4] *It is well known that the problem of* determining validity of quantified Boolean formulas (QBF) *has a complete subproblem for every level of PH. Consider a quantified Boolean formula* $\varphi := \exists \bar{r} \forall \bar{p}_1 \exists \bar{q}_1 \ldots \forall \bar{p}_n \exists \bar{q}_n . \psi$*,* $n \geq 0$*, with quantifier-free* ψ*, which is a Boolean combination of propositional variables from the set* $\bar{r} \cup \bar{p}_1 \cup \ldots \cup \bar{p}_n \cup \bar{q}_1 \cup \ldots \cup \bar{q}_n$*. All valid formulas of this shape together form a complete problem for the* $(2n+1)$*-st level of PH. When we remove the trailing existential quantifier block, we obtain a problem that is complete for the* $(2n)$*-th level.*

PH

A weaker form of Theorem 5.0.3, based on the number of quantifier alternations allowed in SF or GBSR sentences rather than the admitted degrees of variable interaction, leads to a similar pattern of increasing computational difficulty, yet based on a different hierarchy of complexity classes. More concretely, for every positive k *the set of satisfiable SF-sentences restricted to the quantifier prefix* $\exists^*(\forall^*\exists^*)^k$ *gives rise to a* k*-*NExpTime*-complete problem, cf. Figure 5.2 on page 149. The same applies to GBSR. This is easy to see, as every SF or GBSR sentence with at most* n $\forall\exists$ *quantifier alternations has a degree of at most* n*. On the other hand, the SF sentences used in Section 5.3 to encode* k*-*NExpTime*-hard decision problems need exactly* n $\forall\exists$ *quantifier alternations.*

We have already demonstrated that an analysis of the computational complexity of satisfiability problems can greatly benefit from an analysis of how variables co-occur in atoms instead of exclusively considering the number of occurring quantifier alternations. One instance was the derivation of NExpTime*-completeness of MFO-Sat by virtue of Theorem 5.0.3. The reason is simply that the degree of variable interaction might be considerably lower than the number of quantifier alternations.*

What we have not yet taken into account is the Boolean structure of sentences. This may widen the scope of our methods considerably and may moreover help understand where the hardness of satisfiability problems stems from. For example, consider a quantified Boolean formula $\varphi := \forall \bar{p}_1 \exists \bar{q}_1 \ldots \forall \bar{p}_n \exists \bar{q}_n . \psi$ *with quantifier-free* ψ*. As already indicated above, the validity problem for such formulas is complete for the* $(2n+1)$*-st level of PH. But what if, say,* ψ *has the form* $\left(\bigwedge_i K_i\right) \wedge \left(\bigvee_j L_j\right)$*, where the* K_i *and the* L_j *are literals and none of the existential variables in* $\bigwedge_i K_i$ *occurs in* $\bigvee_j L_j$*? Since two distinct Boolean variables can never co-occur in any atom,* φ *can be transformed into the equivalent formula* $\exists \bar{q}_1 \ldots \bar{q}_n \forall \bar{p}_1 \ldots \bar{p}_n . \psi$ *by quantifier shifting. Apparently,* φ *belongs to a class of sentences that resides on the second level of the polynomial hierarchy rather than on the* $(2n+1)$*-st. Indeed, propositional variables in quantified Boolean formulas are as separated as first-order variables are in MFO sentences. It is, hence, easy to see that every quantified Boolean formula can be converted into an equivalent* $\exists^*\forall^*$ *formula. However, this might come at the cost of a super-polynomial blowup in the length of the formula, like we have seen for the MFO case in Theorem 3.2.7. If no such blowup were to occur necessarily, this would indicate a collapse of some levels of PH. Still, it might be worth to reconsider some of the definitions that are based on the shape of quantifier prefixes alone. The ideas sketched in Section 3.6 might also give a good starting point for this endeavor.*

5.1 Computational Complexity of the Existential Fragments of SF and GBSR

A special case that is worth considering are the $\exists^*$ subfragments of SF and GBSR. It is easy to see that these two classes are the same and coincide with the *existential fragment of relational*

[3] This was suggested to the author by Ian Pratt-Hartmann during a discussion at the Seventeenth International Workshop on Logic and Computational Complexity (LCC'16) in Marseille, France, in September 2016.

[4] The exponential-time hierarchy should not be confused with the hierarchy of k-fold (nondeterministic) exponential time for increasing k. The exponential-time hierarchy lies completely within ExpSpace, just like the polynomial-time hierarchy lies entirely within PSpace.

first-order logic. The latter, in turn, is a close relative of propositional logic. We shall refer to this fragment as ∃FO and we shall write Horn-∃FO and Korm-∃FO to address its Horn and Krom subfragments, respectively. As ∃FO without equality is essentially as expressive as propositional logic, it is an easy step to reduce the satisfiability problems associated with equality-free ∃FO and its Horn and Krom subfragments to the corresponding fragments of propositional logic. This even applies to ∃FO and Horn-∃FO with equality. ∃FO

Let *SAT*, *Horn-SAT*, and *Krom-SAT* denote the satisfiability problems associated with the sets of all propositional formulas, propositional Horn formulas, and propositional Krom formulas, respectively. In the literature, Krom-SAT is often called 2SAT. Recall the following well-known results.

Proposition 5.1.1. *SAT is NP-complete [Coo71, Lev73], Horn-SAT is P-complete [JL77, Kas86, Pla84], and Krom-SAT is NL-complete [JLL76].*

In the remainder of the present section we intend to show that (i) satisfiability for ∃FO is NP-complete, (ii) satisfiability for Horn-∃FO is P-complete, (iii) satisfiability for Krom-∃FO without equality is NL-complete. Moreover, we shall see that (iv) satisfiability for Krom-∃FO with equality is complete for NP. The proof of (i) – (iii) proceeds by reductions to the corresponding satisfiability problems for propositional logic and back. This is straightforwardly done by Skolemization as long as we consider only ∃FO sentences without equality, see Lemma 5.1.2. If equality is present in the given ∃FO sentence φ, we first Skolemize exhaustively, thus producing φ_{gnd}, which is ground and contains only Skolem constants and no non-constant function symbols. Then we use the standard trick to eliminate the equality predicate $\approx$. We introduce a fresh binary predicate symbol E and replace every equation $c \approx d$ with the atom $E(c, d)$. Moreover, we add the axioms of a congruence relation for E, i.e. reflexivity, symmetry, transitivity, and compatibility with predicates. Of course, we do not use the universally quantified axioms but rather add their ground instances with respect to all the constant symbols that occur in φ_{gnd}. To avoid an exponential blow-up in the case of the axioms regarding compatibility with predicates, we only add the instances that affect non-equational atoms which really occur in φ_{gnd}. The result is called φ'_{gnd}. Let φ_{prop} be the propositional formula that results from φ'_{gnd} by replacing every ground atom A with the propositional variable p_A. We observe that $\mathrm{len}(\varphi_{\mathrm{prop}}) \in \mathcal{O}\big(\mathrm{len}(\varphi)^3\big)$. Moreover, if φ is a Horn formula, then φ_{prop} is Horn. Notice that the outlined elimination of equality does not preserve the Krom property.

Lemma 5.1.2.

(i) Satisfiability for ∃FO *sentences without equality can be decided nondeterministically in* $\mathrm{poly}\big(\mathrm{len}(\varphi)\big)$ *time.*

(ii) Satisfiability for Horn-∃FO *sentences without equality can be decided deterministically in* $\mathrm{poly}\big(\mathrm{len}(\varphi)\big)$ *time.*

(iii) Satisfiability for Krom-∃FO *sentences without equality can be decided nondeterministically using space that is logarithmic in* $\mathrm{len}(\varphi)$.

Proof. We reduce the above satisfiability problems for ∃FO (sub)fragments to the respective satisfiability problems for propositional logic and vice versa.

Let φ be an ∃FO sentence without equality. Skolemization of all its existential quantifiers leads to the equisatisfiable ground sentence φ_{gnd} in which every atom has the shape $P(c_1, \ldots, c_m)$. Let $A_1, \ldots, A_k$ be a complete enumeration of all the atoms — without duplicates — that occur in φ_{gnd}. Let $q_1, \ldots, q_k$ be a list of pairwise distinct propositional variables. We construct the propositional formula φ_{prop} from φ_{gnd} by replacing every atom A_i with q_i. Clearly, any model $\mathcal{A}$ of φ_{gnd} induces a model $\mathcal{B}$ of φ_{prop}: $\mathcal{B} \models q_i$ if and only if $\mathcal{A} \models A_i$. Conversely, any model $\mathcal{B}'$ of φ_{prop} induces a Herbrand model $\mathcal{A}'$ of φ: $\mathcal{A}' \models A_i$ if and only if $\mathcal{B}' \models q_i$. Consequently, deciding satisfiability of φ can be reduced to deciding satisfiability of φ_{prop}. Moreover, we observe the following properties:

(a) $\mathrm{len}(\varphi_{\mathrm{prop}}) \leq \mathrm{len}(\varphi)$ and $\|\varphi_{\mathrm{prop}}\| \leq \|\varphi\|$.

(b) If φ is a Horn formula, then φ_{prop} is Horn.

(c) If φ is a Krom formula, then φ_{prop} is Krom.

Conversely, any propositional formula over the propositional variables $q_1, \ldots, q_k$ can be straightforwardly transformed into an equisatisfiable ∃FO sentence $\exists y.\, \psi$ with quantifier-free ψ over the monadic atoms $Q_1(y), \ldots, Q_k(y)$. □

Lemma 5.1.3. *There is an effective translation T from* ∃FO *with equality to* ∃FO *without equality such that for every* ∃FO *sentence φ we have that*

(a) every model $\mathcal{A}$ of φ contains a substructure that can be extended to a model $\mathcal{B}$ of $\varphi \wedge T(\varphi)$ over the same domain,

(b) from any model $\mathcal{B}$ of $T(\varphi)$ we can construct a model $\mathcal{A}$ of φ whose domain contains at most as many elements as $\mathcal{B}$'s domain does,

(c) $\text{len}(T(\varphi)) \in \mathcal{O}\big(\text{len}(\varphi)^3\big)$ and $T(\varphi)$ can be computed deterministically in polynomial time,

(d) if φ is Horn, then $T(\varphi)$ is Horn as well.

Proof. We describe the translation T informally. Let φ be some ∃FO sentence with equality. Let φ_{Sk} be the result of Skolemizing all existential quantifiers in φ. Notice that φ_{Sk} does not contain any non-constant function symbols. Let E be a binary predicate symbol that does not occur in φ_{Sk} and let Ω be the set of all constant symbols occurring in φ_{Sk}. We construct the following ground formulas:

$$\psi_{\text{refl}} := \bigwedge_{c \in \Omega} E(c,c) \ ,$$

$$\psi_{\text{symm}} := \bigwedge_{c,d \in \Omega} \big(E(c,d) \rightarrow E(d,c)\big) \ ,$$

$$\psi_{\text{trans}} := \bigwedge_{c,d,e \in \Omega} \big(E(c,d) \wedge E(d,e) \rightarrow E(c,e)\big) \ .$$

Let ψ_{cong} be the conjunction of all ground formulas of the form $E(c_1,d_1) \wedge \ldots \wedge E(c_m,d_m) \wedge P(c_1,\ldots,c_m) \rightarrow P(d_1,\ldots,d_m)$ where $c_1, d_1, \ldots, c_m, d_m \in \Omega$ and P is an m-ary predicate symbol in φ_{Sk}. We write ψ'_{cong} to denote the restriction of ψ_{cong} to formulas $E(c_1,d_1) \wedge \ldots \wedge E(c_m,d_m) \wedge P(c_1,\ldots,c_m) \rightarrow P(d_1,\ldots,d_m)$ whose constituents $P(c_1,\ldots,c_m)$ and $P(d_1,\ldots,d_m)$ actually occur in φ_{Sk} (and are distinct).

Let φ' be the result of replacing every equation $c \approx d$ in φ_{Sk} with the atom $E(c,d)$.

Claim I: φ_{Sk} is satisfiable if and only if $\varphi' \wedge \psi_{\text{refl}} \wedge \psi_{\text{symm}} \wedge \psi_{\text{trans}} \wedge \psi_{\text{cong}}$ is satisfiable.

Proof: Let $\mathcal{A}$ be any model of φ_{Sk}. By the Substructure Lemma, we may assume that $\mathcal{A}$'s domain is $\big\{c^{\mathcal{A}} \bigm| c \in \Omega\big\}$. We now construct a model $\mathcal{B} \models \varphi' \wedge \psi_{\text{refl}} \wedge \psi_{\text{symm}} \wedge \psi_{\text{trans}} \wedge \psi_{\text{cong}}$ from $\mathcal{A}$. We take over $\mathcal{A}$'s domain and its interpretation of the predicate symbols and constant symbols. We define E's interpretation under $\mathcal{B}$ such that $E^{\mathcal{B}} := \big\{\langle \mathsf{a}, \mathsf{a}\rangle \bigm| \mathsf{a} \in \mathsf{A}\big\}$. Hence, for all $c, d \in \Omega$ we observe $\mathcal{B} \models E(c,d)$ if and only if $\mathcal{B} \models c \approx d$. Consequently, $\mathcal{B}$ must be a model of φ_{Sk} and also of $\varphi' \wedge \psi_{\text{refl}} \wedge \psi_{\text{symm}} \wedge \psi_{\text{trans}} \wedge \psi_{\text{cong}}$.

Let $\mathcal{B}$ be a model of $\varphi' \wedge \psi_{\text{refl}} \wedge \psi_{\text{symm}} \wedge \psi_{\text{trans}} \wedge \psi_{\text{cong}}$. By the Substructure Lemma, we may assume that $\mathcal{B}$'s domain is $\mathsf{B} = \big\{c^{\mathcal{B}} \bigm| c \in \Omega\big\}$. We now construct a model $\mathcal{A} \models \varphi_{\text{Sk}}$ from $\mathcal{B}$. Because of $\mathcal{B} \models \psi_{\text{refl}} \wedge \psi_{\text{symm}} \wedge \psi_{\text{trans}} \wedge \psi_{\text{cong}}$, we know that $E^{\mathcal{B}}$ is a congruence relation over B. We define the domain of $\mathcal{A}$ to be the quotient set $\mathsf{A} := \mathsf{B}/_{E^{\mathcal{B}}}$. Moreover, we define $c^{\mathcal{A}} := [c^{\mathcal{B}}]_{E^{\mathcal{B}}}$ for every $c \in \Omega$. For every congruence class $[\mathsf{a}]_{E^{\mathcal{B}}}$ we know that two domain elements $d^{\mathcal{B}}, e^{\mathcal{B}} \in [\mathsf{a}]_{E^{\mathcal{B}}}$ are indistinguishable by the relations $P^{\mathcal{B}}$ for which P occurs in φ'. Therefore, we can use the following definition for every m-ary predicate symbol P in φ' (including E): $P^{\mathcal{A}} := \big\{\big\langle [\mathsf{a}_1]_{E^{\mathcal{B}}}, \ldots, [\mathsf{a}_m]_{E^{\mathcal{B}}}\big\rangle \bigm| \langle \mathsf{a}_1, \ldots, \mathsf{a}_m\rangle \in P^{\mathcal{B}}\big\}$. This yields $\mathcal{A} \models \varphi'$ and for all $c, d \in \Omega$ we observe $\mathcal{A} \models E(c,d)$ if and only if $\mathcal{A} \models c \approx d$. Hence, $\mathcal{A} \models \varphi_{\text{Sk}}$. ◊

It now remains to show equisatisfiability of $\varphi' \wedge \psi_{\text{refl}} \wedge \psi_{\text{symm}} \wedge \psi_{\text{trans}} \wedge \psi_{\text{cong}}$ and $\varphi' \wedge \psi_{\text{refl}} \wedge \psi_{\text{symm}} \wedge \psi_{\text{trans}} \wedge \psi'_{\text{cong}}$. The direction from left to right is obvious.

Claim II: Any model $\mathcal{B} \models \varphi' \wedge \psi_{\text{refl}} \wedge \psi_{\text{symm}} \wedge \psi_{\text{trans}} \wedge \psi'_{\text{cong}}$ gives rise to a model $\mathcal{A} \models \varphi' \wedge \psi_{\text{refl}} \wedge \psi_{\text{symm}} \wedge \psi_{\text{trans}} \wedge \psi_{\text{cong}}$.

Proof: Again, by the Substructure Lemma, we may assume that $\mathcal{B}$'s domain is $\{c^{\mathcal{B}} \mid c \in \Omega\}$. Let $P(c_1, \ldots, c_m)$ and $P(d_1, \ldots, d_m)$ be two atoms that occur in φ'. We observe the following property:

(∗) If $E^{\mathcal{B}}$ contains the pairs $\langle c_1^{\mathcal{B}}, d_1^{\mathcal{B}}\rangle, \ldots, \langle c_m^{\mathcal{B}}, d_m^{\mathcal{B}}\rangle$, then $\mathcal{B} \models \psi'_{\text{cong}}$ entails that $\mathcal{B} \models P(c_1, \ldots, c_m)$ holds if and only if $\mathcal{B} \models P(d_1, \ldots, d_m)$ does.

We define $\mathcal{A}$ such that $\mathsf{A} := \mathsf{B}$, $E^{\mathcal{A}} := E^{\mathcal{B}}$, and $c^{\mathcal{A}} := c^{\mathcal{B}}$ for every $c \in \Omega$. Moreover, for every m-ary predicate symbol P occurring in φ' and every tuple $\langle \mathsf{a}_1, \ldots, \mathsf{a}_m\rangle \in \mathsf{A}$ we set $\langle \mathsf{a}_1, \ldots, \mathsf{a}_m\rangle \in P^{\mathcal{A}}$ if and only if there is some atom $P(c_1, \ldots, c_m)$ in φ' for which we have $\langle c_1^{\mathcal{A}}, \mathsf{a}_1\rangle, \ldots, \langle c_m^{\mathcal{A}}, \mathsf{a}_m\rangle \in E^{\mathcal{A}}$ and $\mathcal{B} \models P(c_1, \ldots, c_m)$. Due to (∗), we know that $\mathcal{A} \models \varphi'$ still holds. By construction of $\mathcal{A}$, we moreover observe $\mathcal{A} \models \psi_{\text{refl}} \wedge \psi_{\text{symm}} \wedge \psi_{\text{trans}} \wedge \psi_{\text{cong}}$. ◊

We set $T(\varphi) := \varphi' \wedge \psi_{\text{refl}} \wedge \psi_{\text{symm}} \wedge \psi_{\text{trans}} \wedge \psi'_{\text{cong}}$ for every ∃FO sentence φ. It is easy to see that if φ is Horn then $T(\varphi)$ is Horn as well. Moreover, the length of $T(\varphi)$ is bounded from above by $k \cdot \big(\text{len}(\varphi) + |\Omega|^3 + \text{len}(\varphi) \cdot |\text{At}(\varphi_{\text{Sk}})|^2\big)$ for some positive integer k, where $\text{At}(\varphi_{\text{Sk}})$ denotes the set of all non-equational atoms that occur in φ_{Sk}. □

Proposition 5.1.4.

(i) Satisfiability for ∃FO *is NP-complete.*

(ii) Satisfiability for Horn-∃FO *is P-complete.*

(iii) Satisfiability for Krom-∃FO *without equality is NL-complete.*

(iv) Satisfiability for Krom-∃FO *with equality is NP-complete.*

Proof sketch. The membership in the respective complexity classes is settled in Lemmas 5.1.2 and 5.1.3 for (i) – (iii). The membership part of (iv) follows from (i). In order to show hardness for these cases, it remains to reduce the respective SAT problems to the corresponding satisfiability problems for SF.

Let φ_{prop} be some propositional sentence and let $q_1, \ldots, q_k$ be a complete list of all propositional variables occurring in φ_{prop} (without duplicates). Let $y_1, \ldots, y_k$ be pairwise distinct first-order variables. We construct the first-order sentence $\exists \bar{y}.\, \varphi$ from φ_{prop} by replacing every q_i by the atom $P(y_i)$ and adding the quantifier block $\exists y_1 \ldots y_k$ to the front. By similar arguments as we have used in the proof of Lemma 5.1.2, we can show that φ_{prop} is satisfiable if and only if φ is satisfiable. Moreover, we observe the following properties:

(a) $\text{len}(\varphi) \in \mathcal{O}\big(\text{len}(\varphi_{\text{prop}})\big)$ and φ is computable deterministically in polynomial time.

(b) If φ_{prop} is a Horn formula, then φ is Horn.

(c) If φ_{prop} is a Krom formula, then φ is Krom.

The outlined construction polynomially reduces SAT, Horn-SAT, and Krom-SAT to the respective satisfiability problems for ∃FO, Horn-∃FO, and Krom-∃FO, all without equality.

It remains to show the NP-hardness part of (iv). We reduce 3SAT — the satisfiability problem for propositional formulas in CNF in which each clause contains at most three literals —, which was shown to be NP-hard by Cook [Coo71].[5] Let $\varphi := \varphi_3 \wedge \varphi_{\leq 2}$ be any propositional formula in conjunctive normal form where φ_3 is a conjunction of clauses that contain exactly three literals each

[5]The idea underlying the reduction was suggested to the author of the present thesis by Christoph Weidenbach during a discussion in October 2018.

and $\varphi_{\leq 2}$ is a conjunction of clauses with at most two literals each. Our first step will be to transform φ into an equisatisfiable formula such that all clauses with three literals are Horn and do not share any propositional variables. To achieve this goal, we introduce fresh propositional variables together with definitions of the form $p \leftrightarrow q$ or $p \leftrightarrow \neg q$, each of which can readily be transformed into two two-literal clauses, respectively. Put more precisely, we construct two propositional formulas φ_3' and φ_{def} such that

(d) φ_{def} is a conjunction of formulas of the form $p \leftrightarrow q$ or $p \leftrightarrow \neg q$ with $p \neq q$;

(e) φ_3' is a conjunction of pairwise variable-disjoint Horn clauses;

(f) every clause in φ_3' contains exactly three literals, exactly one positive literal, and exactly three pairwise distinct propositional variables;

(g) none of the propositional variables in φ_3' occurs in $\varphi_{\leq 2}$ and each of the variables in φ_3' occurs in exactly one clause in φ_{def},

(h) we have $\varphi_3' \wedge \varphi_{\text{def}} \wedge \varphi_{\leq 2} \models \varphi_3 \wedge \varphi_{\leq 2}$ and every satisfying assignment for $\varphi_3 \wedge \varphi_{\leq 2}$ can be extended to some satisfying assignment for $\varphi_3' \wedge \varphi_{\text{def}} \wedge \varphi_{\leq 2}$.

Then, we have $\varphi_3' = \bigwedge_{1 \leq i \leq m} (\neg p_i \vee \neg q_i \vee r_i)$ for some integer m and pairwise-distinct propositional variables $p_1, \ldots, p_m, q_1, \ldots, q_m, r_1, \ldots, r_m$. Let φ_3'' be the first-order formula

$$\varphi_3'' := \bigwedge_{1 \leq i \leq m} (a_i \approx b_i \wedge b_i \approx c_i \rightarrow a_i \approx c_i)$$

such that all $a_1, \ldots, a_m, b_1, \ldots, b_m, c_1, \ldots, c_m$ are pairwise-distinct constant symbols. Moreover, let φ_{def}' result from φ_{def} after the following transformations. We replace every atom p_i with $a_i \approx b_i$, every atom q_i with $b_i \approx c_i$, and every atom r_i with $a_i \approx c_i$. In addition, we replace every propositional variable q that occurs in φ_{def} but not in φ_3' with the first-order atom $P_q(d)$ for some fresh predicate symbol P_q and some constant symbol d that is different from all the a_i, b_i, c_i. Let $\varphi_{\leq 2}'$ be the result of repeating the latter replacement for $\varphi_{\leq 2}$. Then, the $\exists$FO sentence $\exists a_1 \ldots a_m b_1 \ldots b_m c_1 \ldots c_m d.\, \varphi_3'' \wedge \varphi_{\text{def}}' \wedge \varphi_{\leq 2}'$ is satisfiable if and only if $\varphi_3' \wedge \varphi_{\text{def}} \wedge \varphi_{\leq 2}$ is satisfiable. Finally, notice that every clause in φ_3'' is an instance of the transitivity axiom of equality and, hence, it is a tautology. Consequently, $\varphi_3 \wedge \varphi_{\text{def}} \wedge \varphi_{\leq 2}$ is satisfiable if and only if $\varphi' := \exists a_1 \ldots a_m b_1 \ldots b_m c_1 \ldots c_m d.\, \varphi_{\text{def}}' \wedge \varphi_{\leq 2}'$ is satisfiable, and the latter sentence is equivalent to some Krom-$\exists$FO sentence whose length is linear in $\text{len}(\varphi')$. Moreover, $\text{len}(\varphi')$ is linear in $\text{len}(\varphi)$. Therefore, 3SAT is polynomially reducible to the satisfiability problem for Krom-$\exists$FO sentences with equality. □

5.2 Horn and Krom Special Cases of SF and a Conjecture

It is well known that the restriction to Horn or Krom sentences can tremendously reduce the computational effort required to decide satisfiability of first-order formulas, unless widely believed conjectures in computational complexity theory fail to be true. We have already recalled in Proposition 5.1.1 that this holds true when going from SAT to Horn-SAT or Krom-SAT. But the effect is also known for Horn and Krom variants of BS-Sat and MFO-Sat. Table 5.1 provides an overview of satisfiability problems that are complete for the complexity classes NL, P, NP, PSPACE, EXPTIME, and k-NEXPTIME for $k \geq 1$. When we understand $\exists^*$-sentences without equality — after exhaustive Skolemization — as being essentially equivalent to propositional sentences — cf. Lemma 5.1.2 and its proof —, then we can conceive SAT as a restricted case of SF-Sat, namely $\exists^*$-SF-Sat (= $\exists$FO-Sat). Following this train of thought further leads to the correspondence between Horn-SAT and Horn-$\exists^*$-SF-Sat (= Horn-$\exists$FO-Sat) and the correspondence between Krom-SAT and Krom-$\exists^*$-SF-Sat (=Krom-$\exists$FO-Sat).

Given the complexity hierarchies that result from Theorem 5.0.3 together with the fact that the number of $\forall\exists$-alternations in an SF sentence φ bounds the degree $\partial_\exists(\varphi)$ from above, we conclude that

Classes	Complete problems	References
NL	Krom-SAT	[JLL76]
P	Horn-SAT	[JL77, Kas86, Pla84]
	Krom-MFO-Sat	[DL84a]
NP	SAT	[Coo71, Lev73]
PSpace	Krom-BS-Sat	[DL84a, Pla84]
	Krom-BSR-Sat	Proposition 5.2.1
ExpTime	Horn-BS-Sat	[CLM81, DL84a, Pla84]
	Horn-BSR-Sat	Proposition 5.2.1
	Horn-MFO-Sat	[DL84b], see also Proposition 3.14.7
	Maslov-Sat	[DL84a]
NExpTime	BSR-Sat	[Lew80]
k-NExpTime	$\mathrm{SF}_{\leq k}$-Sat and $\mathrm{GBSR}_{\leq k}$-Sat	Theorem 5.0.3

Table 5.1: Some basic complexity classes and corresponding complete problems. Recall that BS stands for the Bernays–Schönfinkel fragment, i.e. BSR without equality. Maslov-Sat denotes the satisfiability problem associated with the Maslov fragment, i.e. the set of satisfiable relational $\exists^*\forall^*\exists^*$ sentences without equality that are Krom, cf. page 26. Except for Maslov-Sat, all mentioned problems can be conceived as special cases of SF-Sat and GBSR-Sat. The lower-bound proof for the Maslov fragment in [DL84a] is based on a $\forall^*\exists$-sentence that is neither in SF nor in GBSR.

SF-Sat restricted to sentences with a $\exists^*(\forall^*\exists^*)^k$ quantifier prefix yields a k-NExpTime-complete problem, where 0-NExpTime is understood to be NP. With the results from Table 1, we already know the computational complexity of certain special cases for SF-Sat when restricted to $\exists^*$ and $\exists^*\forall^*$ quantifier prefixes and Krom or Horn form without equality. The following proposition shows that also in the case with equality the restriction to Horn or Krom sentences yields computationally less hard satisfiability problem, unless NExpTime equals ExpTime and/or PSpace.

Proposition 5.2.1. *Horn-BSR-Sat is in* ExpTime *and Krom-BSR-Sat is in* PSpace.

Proof sketch. The reduction of Horn-BSR-Sat to Horn-BS-Sat is even simpler than the reduction of Horn-$\exists$FO-Sat with equality to Horn-$\exists$FO-Sat without equality. We again replace $\approx$ with a fresh binary predicate E and conjoin the equality axioms for E to the original formula: reflexivity, symmetry, transitivity, and compatibility with predicates. As universal quantification is available now, this can be done using BS formulas whose length is polynomial in the underlying vocabulary.

We now turn our attention to the case of Krom BSR sentences. Usually, upper bounds for Krom fragments are shown via a chain argument (cf. [DL84a], and Section 8.3.1 in [BGG97]). We recap the idea for Krom-BS-Sat before we use it for the case with equality. Let ψ be a Krom BS sentence without equality after exhaustive Skolemization. We assume that ψ has the shape $\forall\bar{\mathrm{x}}.\ \bigwedge_{i\in I}(L_i \vee K_i)$ where the L_i and K_i are literals. This is not a restriction, as every unit clause L is equivalent to $L \vee L$. We denote by $\mathcal{G}(\psi)$ the directed graph with the following components: The vertex set V is the set of all ground literals that can be built from the predicate and constant symbols in ψ. The edge relation E contains an edge from L to K if and only if ψ contains a clause that has an instance equivalent to $L \to K$. It can be shown that ψ is unsatisfiable if and only if the graph $\mathcal{G}(\psi)$ contains a cycle along which some ground atom A and its negation occur (Lemmas 8.3.1, 8.3.4, and 8.3.5 in [BGG97]).

$\mathcal{G}(\psi) = \langle V, E\rangle$

A PSpace decision procedure for Krom-BS-Sat then detects such a cycle by nondeterministically choosing the ground atom A as its starting point and then, again nondeterministically, exploring a shortest path to $\neg A$ in $\mathcal{G}(\psi)$ by constructing the appropriate instances of clauses $L \vee K$ nondeterministically one after the other. Notice that such a shortest path is not longer than $|V| \leq \mathrm{len}(\varphi) \cdot 2^{\log(\mathrm{len}(\varphi))\cdot \mathrm{len}(\varphi)}$, which yields a termination criterion for the nondeterministic procedure. During that process, the procedure at every step only needs to store the instance A, (an instance of) the current clause $L \vee K$, including the direction of the considered edge from $\mathcal{G}(\psi)$,

and the length of the currently explored path. If a path from A to $\neg A$ has been detected, the procedure starts to seek for a path from $\neg A$ to A. If it succeeds in both directions, the sentence ψ is unsatisfiable. Conversely, the procedure succeeds whenever ψ is unsatisfiable. This entails that checking unsatisfiability for Krom BS sentences is in NPSPACE and, hence, that Krom-BS-Sat is in co-NPSPACE. But since the two complexity classes coincide with PSPACE, which is a consequence of the famous theorems by Savitch [Sav70] and Immerman and Szelepcsényi [Imm88, Sze88], it follows that both problems are in PSPACE.

φ'

We now extend this idea so that equality can be handled. Consider any Krom BSR sentence $\varphi := \exists\bar{\mathrm{z}}\forall\bar{\mathrm{x}}.\,\chi$ with quantifier-free χ. As a first step, we construct the Krom BSR sentence φ' from φ as follows. Consider each and every clause C in φ and let $\{x_1, x_2, x_3, x_4\} \subseteq \bar{\mathrm{x}}$ be the set of all universally quantified variables that occur in equations in C. Since C is a Krom clause, this set contains at most four variables. We replace C with the conjunction $\bigwedge_{z_1,z_2,z_3,z_4\in\bar{\mathrm{z}}} C\big[x_1/z_1, \ldots, x_4/z_4\big]$. The resulting sentence φ' is a Krom BSR sentence with $\mathrm{len}(\varphi') \leq \big(\mathrm{len}(\varphi)\big)^5$ in which all variables occurring in equations are existentially quantified. Moreover, it is easy to see that $\varphi \models \varphi'$ and that any model $\mathcal{A} \models \varphi'$ contains some substructure that is also a model of φ. Such a substructure of $\mathcal{A}$ is induced by any set $\{\mathsf{a}_i \in \mathsf{A} \mid \mathcal{A}, [\bar{\mathrm{z}} \models \bar{\mathsf{a}}] \models \forall\bar{\mathrm{x}}.\,\chi \text{ and } \bar{\mathsf{a}} = \langle \mathsf{a}_1, \ldots, \mathsf{a}_{|\bar{\mathrm{z}}|}\rangle\}$ — there exists at least one such set if $\mathcal{A}$ is a model of φ'.

$\prec$, S

For the rest of this proof we fix some strict linear order $\prec$ on the variables in $\bar{\mathrm{z}}$ and define S to be the set $\{z \approx z' \mid z, z' \in \bar{\mathrm{z}}\}$. Every subset $T \subseteq S$ induces an equivalence relation $\sim_T$

$\sim_T$

over the variables in $\bar{\mathrm{z}}$. Although there are $2^{|\bar{\mathrm{z}}|^2}$ such subsets T, there are only $B_{|\bar{\mathrm{z}}|}$ equivalence classes over $|\bar{\mathrm{z}}|$, where B_n denotes the n-th *Bell number*, which is known to be bounded by $B_n < \big(\frac{0.792n}{\ln(n+1)}\big)^n$ (Theorem 2.1 in [BT10]). Operationally, such an equivalence relation can be

ρ_T

represented by a *rewrite function* $\rho_T : \bar{\mathrm{z}} \to \bar{\mathrm{z}}$ that maps any $z \in \bar{\mathrm{z}}$ to the least $z' \sim_T z$ (with respect to $\prec$). Storing ρ_T requires polynomial space only.

The nondeterministic decision procedure for Krom BSR sentences employs the one described above for Krom BS sentences as subroutine. Given any Krom BSR sentence φ, the procedure transforms it into some equisatisfiable sentence φ' in which all variables in equations are existentially quantified, just as described above. Then, the procedure iterates over all rewrite functions ρ_T

φ'_T

one after another, and considers the sentence φ'_T that is a copy of $\varphi' = \exists\bar{\mathrm{z}}\forall\bar{\mathrm{x}}.\,\chi'$ in which every $z \in \bar{\mathrm{z}}$ in χ' is replaced with the variable $\rho(z)$. In case of $\rho(z) \neq z$, the quantifier $\exists z$ is removed from the quantifier prefix. After these replacements, every trivial equation $z \approx z$ is replaced with the logical constant `true`, and every equation $z \approx z'$ with $z \neq z'$ is replaced with `false`. For technical reasons the procedure afterwards removes all logical constants `true` and `false` in a way that preserves semantics and so that in the resulting (equivalent) sentence every clause consists of two non-equational literals. Hence, φ'_T is a Krom BS sentence. Now the procedure employs the subroutine described above to check whether φ'_T is unsatisfiable or not, using polynomial space only. If it is, the procedure deletes φ'_T from memory — recall that the original φ' was kept — and goes on iterating over the rewrite functions ρ_T. If φ'_T happens to be satisfiable, then the original φ is satisfiable as well. If all the φ'_T with $T \subseteq S$ are found to be unsatisfiable, then the original φ is unsatisfiable. To keep track of the iteration over all $T \subseteq S$, a counter suffices that can be represented with $|\bar{\mathrm{z}}|^2$ bits. This shows that Krom-BSR-Sat is in PSPACE. □

It is tempting to try to extrapolate a pattern from the observations in Table 5.1, Proposition 5.2.1, and Theorem 5.0.3, and speculate that $\exists^*(\forall^*\exists^*)^k$-Horn-SF-Sat is complete for k-EXPTIME for every $k \geq 1$, and that $\exists^*(\forall^*\exists^*)^k$-Krom-SF-Sat is complete for $(k-1)$-EXPSPACE for every $k \geq 2$. The overall picture of this conjecture is sketched in Figure 5.2. The conjecture can also be formulated in a more precise way in terms of the degree $\partial_\exists(\varphi)$ of SF sentences φ, i.e. for Horn-SF$_{\leq k}$-Sat and Krom-SF$_{\leq k}$-Sat. Then, the EXPTIME-completeness of Horn-MFO-Sat fits nicely into the picture, as MFO sentences have degree at most one. The membership of Krom-MFO-Sat in P indicates that this problem is very likely to be easier to solve than what one would have gotten from the extrapolation.

Conjecture 5.2.2. *For every positive k we conjecture that Horn-*SF$_{\leq k}$*-*Sat *is complete for k-*EXPTIME*. Moreover, we conjecture that Krom-*SF$_{\leq k}$*-*Sat *is complete for $(k-1)$-*EXPSPACE*, where*

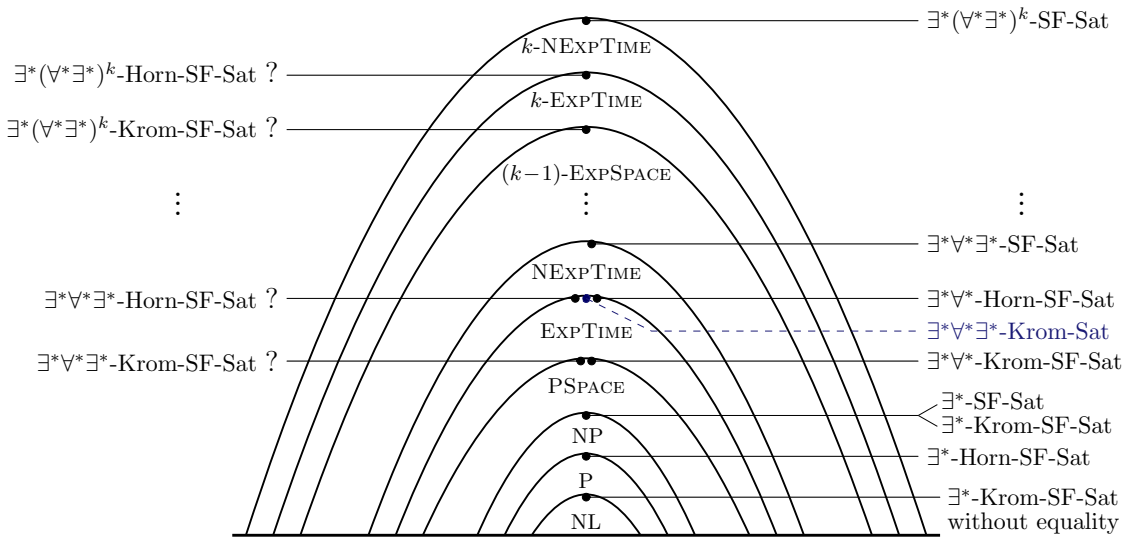


Figure 5.2: On the left-hand side the conjectured pattern of complexity is shown for the sets of $\exists^*(\forall^*\exists^*)^k$-SF-sentences that are Horn and Krom, respectively. On the right-hand side the known complexity results are depicted. Note that the blue part stems from the complexity of the Maslov fragment.

0-ExpSpace *is meant as a synonym for* PSpace.

Another hint that may speak in favor of the conjecture that also the satisfiability problems for Krom and Horn SF sentences become harder with an increasing number of quantifier alternations is given by Theorem 3.2.7. In that theorem we have shown the existence of an SF sentence φ_n, for arbitrary $n \geq 1$, that is Krom and Horn and whose shortest BSR equivalent has a length that is n-fold exponential in $\text{len}(\varphi_n)$. Of course, all of the above said also applies when we replace SF with GBSR. Hence, the same conjecture can be made for Horn-GBSR$_{\leq k}$-Sat and Krom-GBSR$_{\leq k}$-Sat.

5.3 Proving Lower Bounds for SF-Sat

In the present section we establish lower bounds regarding the worst-case time complexity of SF-Sat. Our arguments will be based on a particular form of bounded domino (or tiling) problems developed by Grädel (see [Grä90a] and [BGG97], Section 6.1.1). By $\mathbb{Z}_t$ we denote the set of integers $\{0, \ldots, t-1\}$ for any positive $t \geq 1$.

Definition 5.3.1 (Bounded domino systems, cf. Definition 6.1.1 in [BGG97])**.** *A* domino system $\mathfrak{D} := \langle \mathcal{D}, \mathcal{H}, \mathcal{V} \rangle$ *is a triple where* $\mathcal{D}$ *is a finite set of tiles and* $\mathcal{H}, \mathcal{V} \subseteq \mathcal{D} \times \mathcal{D}$ *are binary relations determining the allowed horizontal and vertical neighbors of tiles, respectively. Consider the torus* $\mathbb{Z}_t^2 := \mathbb{Z}_t \times \mathbb{Z}_t$ *and let* $\overline{D} := D_0 \ldots D_{n-1}$ *be a word over* $\mathcal{D}$ *of length* $n \leq t$*. The letters of* $\overline{D}$ *represent tiles. We say that* $\mathfrak{D}$ tiles the torus $\mathbb{Z}_t^2$ with initial condition $\overline{D}$ *if and only if there exists a mapping* $\tau : \mathbb{Z}_t^2 \to \mathcal{D}$ *such that for every* $\langle x, y \rangle \in \mathbb{Z}_t^2$ *the following conditions hold, where "+1" denotes increment modulo* t*.* $\mathfrak{D} = \langle \mathcal{D}, \mathcal{H}, \mathcal{V} \rangle$

(a) *If* $\tau(x, y) = D$ *and* $\tau(x+1, y) = D'$*, then* $\langle D, D' \rangle \in \mathcal{H}$*.*

(b) *If* $\tau(x, y) = D$ *and* $\tau(x, y+1) = D'$*, then* $\langle D, D' \rangle \in \mathcal{V}$*.*

(c) $\tau(i, 0) = D_i$ *for* $i = 0, \ldots, n-1$*.*

Notice that Definition 5.3.1 is similar to Definition 3.3.1 on page 41, where we introduced unconstrained domino problems. The main difference lies in the fact that the bounded variant of domino systems speaks about tiling a torus, i.e. a finite space, in contrast to an infinite plane.

Definition 5.3.2 (Bounded domino problems, cf. Definition 6.1.5 in [BGG97]). *Let $T : \mathbb{N} \to \mathbb{N}$ be a function representing some time bound and let $\mathfrak{D} := \langle \mathcal{D}, \mathcal{H}, \mathcal{V} \rangle$ be a domino system. The problem* DOMINO $\mathrm{DOMINO}(\mathfrak{D}, T(n))$ *is the set of those words $\overline{D}$ over the alphabet $\mathcal{D}$ for which $\mathfrak{D}$ tiles $\mathbb{Z}^2_{T(|\overline{D}|)}$ with initial condition $\overline{D}$.*

Bounded domino problems provide a convenient way of deriving lower bounds via reductions. Suppose we are given some well-behaved time bound $T(n)$ that grows sufficiently fast. Further assume there is a reasonable translation from $\mathrm{DOMINO}(\mathfrak{D}, T(n))$ into some problem $\mathcal{L}$ where the length of the results is bounded from above by some function $g(n)$. It then follows that the time required to solve the hardest instances of $\mathcal{L}$ lies in $\Omega\big(T(h(n))\big)$, where $h(n)$ may be conceived as an inverse of $g(n)$ from an asymptotic point of view. Proposition 5.3.6 shall formalize this observation. But before we write it down, we need some more results to establish the link between resource-bounded Turing machines and bounded domino problems. Moreover, in order to derive hardness results for (subproblems of) SF-Sat via the reductions from bounded domino problems, we need more knowledge about the computational hardness of these problems.

Proposition 5.3.3 ([BGG97], Theorem 6.1.2). *Let M be a* simple *nondeterministic one-tape Turing machine with input alphabet Γ. Then there is a domino system $\mathfrak{D} = \langle \mathcal{D}, \mathcal{H}, \mathcal{V} \rangle$ and a linear-time reduction which takes any input $w \in \Gamma^*$ to some word $\overline{D} \in \mathcal{D}^*$ with $|w| = |\overline{D}|$ such that*

(a) if M accepts w in time t_0 with space s_0, then $\mathfrak{D}$ tiles $\mathbb{Z}_s \times \mathbb{Z}_t$ with initial condition $\overline{D}$ for all $s \geq s_0 + 2$ and $t \geq t_0 + 2$;

(b) if M does not accept w, then $\mathfrak{D}$ does not tile $\mathbb{Z}_s \times \mathbb{Z}_t$ with initial condition $\overline{D}$ for any $s, t \geq 2$.

simple Turing machines

By a *simple* Turing machine the authors of [BGG97] mean a nondeterministic one-tape Turing machine M over the input alphabet Γ that meets the following conditions:

> "The alphabet of M contains Γ and at least one other symbol $\square$ (blank). M works on a semi-infinite tape and never tries to move left from the left-most tape cell. At every stage of the computation there is some s such the tape cells $0, \ldots, s$ contain only non-blank symbols, all other tape cells contain $\square$; in particular, to the right of a blank only other blanks may appear. Furthermore, we assume that M has a unique accepting configuration: the machine is in the unique accepting state q_a, the tape contains only blanks and the head is in position 0.
>
> These conditions do not restrict computational power. Every language accepted in time $T(n)$ and space $S(n)$ by some one-tape nondeterministic Turing machine is accepted within the same time and space bounds by a simple Turing machine, as long as $S(n), T(n) \geq 2n$." [BGG97], page 243

Proposition 5.3.4 ([BGG97], Theorem 6.1.6). *We call a function $T : \mathbb{N} \to \mathbb{N}$* time constructible *if there exists a deterministic Turing machine making precisely $T(n)$ steps on inputs of length n.*

Let $T : \mathbb{N} \to \mathbb{N}$ be a time-constructible function with $\big(T(dn)\big)^2 \in o(T(n))$ for some constant $d > 0$. There exists a domino system $\mathfrak{D}$ and a constant $c > 0$ such that $\mathrm{DOMINO}(\mathfrak{D}, T(n)) \notin \mathrm{NTIME}(T(cn))$.

Definition 5.3.5 (Polynomially reducible, cf. Definition 6.1.7 in [BGG97]). *Let Γ, Δ be two alphabets and let $\mathcal{L} \subseteq \Gamma^*$ and $\mathcal{K} \subseteq \Delta^*$ be two problems (formal languages). Moreover, let $g : \mathbb{N} \to \mathbb{N}$ be some function. We say that $\mathcal{L}$* is polynomially reducible to $\mathcal{K}$ via length order $g(n)$, *denoted $\mathcal{L} \leq_{g(n)} \mathcal{K}$, if there exists a total mapping $f : \Gamma^* \to \Delta^*$ which is computable in polynomial time such that for every $w \in \Gamma^*$ we have $|f(w)| \leq \mathcal{O}\big(g(|w|)\big)$ and $f(w) \in \mathcal{K}$ if and only if $w \in \mathcal{L}$.*

Proposition 5.3.6 ([BGG97], Theorem 6.1.8)**.** *Let $T : \mathbb{N} \to \mathbb{N}$ be a time-constructible function with $\big(T(dn)\big)^2 \in o(T(n))$ for some constant $d > 0$ and let $\mathcal{L}$ be a problem such that for every domino system $\mathfrak{D}$ we have $\mathrm{DOMINO}(\mathfrak{D}, T(n)) \leq_{g(n)} \mathcal{L}$, i.e. $\mathrm{DOMINO}(\mathfrak{D}, T(n))$ is* polynomially reducible *to $\mathcal{L}$ via length order $g(n)$. Moreover, let $h : \mathbb{N} \to \mathbb{N}$ be a function such that $h(e \cdot g(n)) \in \mathcal{O}(n)$ for any positive constant e. There exists a positive constant $c > 0$ such that $\mathcal{L} \notin \mathrm{NTIME}(T(c \cdot h(n)))$.*

We are now almost done with the preliminaries. It only remains to show that the functions we intend to use as time bounds $T(n)$ satisfy the requirements of Proposition 5.3.6. Recall our notation for the *tetration operation* $2^{\uparrow k}(m)$, which we defined inductively: $2^{\uparrow 0}(m) := m$ and $2^{\uparrow k+1}(m) := 2^{\left(2^{\uparrow k}(m)\right)}$. In addition, we use the short-hand $2^{\uparrow k}$ to abbreviate $2^{\uparrow k}(2)$. We need $2^{\uparrow k}$ to find positive constants c_1, c_2 and verify the conditions $\big(2^{\uparrow k}(c_1 n)\big)^2 \in o(2^{\uparrow k}(n))$ for every fixed $k \geq 1$, and $\big(2^{\uparrow c_2 n}\big)^2 \in o(2^{\uparrow n})$, respectively. Setting $c_1 := \frac{1}{4}$ and $c_2 := \frac{1}{2}$ entails the following.

Lemma 5.3.7. *Let $k \geq 1$ be some fixed positive integer. For every positive constant $c > 0$ there exists some positive integer $n_0 \geq 1$ such that for every $n \geq n_0$ we have*

$$\Big(2^{\uparrow k}(\lceil n/4 \rceil)\Big)^2 \leq c \cdot 2^{\uparrow k}(n) \ .$$

Proof. Let $k = 1$. We observe

$$\Big(2^{\uparrow 1}(n/4)\Big)^2 = 2^{2 \cdot n/4} = 2^{n/2} \ .$$

If $c \geq 1$, then $2^{n/2} \leq c \cdot 2^n$ is obvious.

Assume $0 < c < 1$ and set $d := \frac{1}{c}$. Hence, $d > 1$. If remains to show $d \cdot 2^{n/2} \leq 2^n$ for every sufficiently large n. Due to $2^n = 2^{n/2+n/2} = 2^{n/2} \cdot 2^{n/2}$, we observe $d \cdot 2^{n/2} \leq 2^{n/2} \cdot 2^{n/2}$ if and only if $d \leq 2^{n/2}$. But the latter certainly holds for sufficiently large n.

Let $k = 2$. If $c \geq 1$, then $2^{\uparrow 2}(n) \leq c \cdot 2^{\uparrow 2}(n)$. It thus suffices to show $2^{2 \cdot 2^{n/4}} \leq 2^{2^n} = 2^{\uparrow 2}(n)$ or, equivalently, $\frac{1}{4}n + 1 \leq n$ for sufficiently large n. But this is obviously true.

Assume $0 < c < 1$ and define $d := \frac{1}{c}$. It follows that $d > 1$. It remains to show $d \cdot 2^{2^{n/4+1}} \leq 2^{2^n}$ for sufficiently large n. Due to $2^{2^n} = 2^{2^{n/2} \cdot 2^{n/2}} \geq 2^{2^{n/2}+2^{n/2}} = 2^{2^{n/2}} \cdot 2^{2^{n/2}}$ with $d \leq 2^{2^{n/2}}$ and $2^{2^{n/4+1}} \leq 2^{2^{n/2}}$ for sufficiently large n, we also observe $d \cdot 2^{2^{n/4+1}} \leq 2^{2^{n/2}} \cdot 2^{2^{n/2}} \leq 2^{2^n}$ for sufficiently large n.

Analogous arguments hold for every $k \geq 2$. □

Lemma 5.3.8. *For every positive constant $c > 0$ there exists some positive integer $n_0 > 0$ such that for every $n \geq n_0$ we have*

$$\Big(2^{\uparrow \lceil n/2 \rceil}\Big)^2 \leq c \cdot 2^{\uparrow n} \ .$$

Proof. We distinguish two cases:

Suppose $c \geq 1$. We observe

(1) $\Big(2^{\uparrow \lceil n/2 \rceil}\Big)^2 = 2^{2 \cdot 2^{2^{\uparrow \lceil n/2 \rceil - 2}}} = 2^{2^{1+2^{\uparrow \lceil n/2 \rceil - 2}}}$, and

(2) $c \cdot 2^{\uparrow n} \geq 2^{\uparrow n} = 2^{2^{2^{\uparrow n-2}}}$.

Hence, it suffices to show there is some $n_0 \geq 2$ such that $1 + 2^{\uparrow \lceil n/2 \rceil - 2} \leq 2^{\uparrow n-2}$ holds for every $n \geq n_0$. One possible choice is $n_0 = 4$

Suppose $0 < c < 1$. We set $d := \frac{1}{c}$. Due to $d > 1$, $\log_2 d$ is defined. Moreover, we observe

$$\begin{aligned} d \cdot \Big(2^{\uparrow \lceil n/2 \rceil}\Big)^2 &= 2^{\log_2 d} \cdot \Big(2^{\uparrow \lceil n/2 \rceil}\Big)^2 &&= 2^{\log_2 d + 2 \cdot 2^{\uparrow \lceil n/2 \rceil - 1}} \\ &\leq 2^{\log_2 d \cdot 2^{\uparrow \lceil n/2 \rceil - 1} + 2 \cdot 2^{\uparrow \lceil n/2 \rceil - 1}} &&= 2^{(\log_2 d + 2) \cdot 2^{\uparrow \lceil n/2 \rceil - 1}} \\ &= 2^{2^{\log_2(\log_2 d + 2)} \cdot 2^{\uparrow \lceil n/2 \rceil - 2}} &&= 2^{2^{\log_2(\log_2 d + 2) + 2^{\uparrow \lceil n/2 \rceil - 2}}} \ . \end{aligned}$$

Hence, in order to prove that there is some n_0 such that $d \cdot \left(2^{\uparrow\lceil n/2\rceil}\right)^2 \leq 2^{\uparrow n}$ holds for every $n \geq n_0$, it suffices to show that there is some n_0 such that

$$d' + 2^{\uparrow\lceil n/2\rceil - 2} \quad \leq \quad 2^{\uparrow n-2}$$

where $d' := \log_2(\log_2 d + 2)$. The proof thus boils down to asking whether the difference $2^{\uparrow n-2} - 2^{\uparrow\lceil n/2\rceil-2}$ exceeds any constant value d' for sufficiently large n. This is certainly the case. □

The following lemma contains the key technical result of the present section: reductions of bounded domino problems to $\mathrm{SF}_{\leq k}$-Sat, for any positive k, and SF-Sat, respectively.

Lemma 5.3.9.

(i) Fix some positive integer $k > 0$ and let $\mathfrak{D}$ be any domino system. Let $\mathrm{Sat}(\mathrm{SF}_{\partial_\exists \leq k})$ be the set containing all satisfiable SF sentences φ whose degree $\partial_\exists(\varphi)$ is at most k. We have $\mathrm{DOMINO}\big(\mathfrak{D}, 2^{\uparrow k}(n)\big) \leq_{n \cdot \log n} \mathrm{Sat}(\mathrm{SF}_{\partial_\exists \leq k})$.

(ii) Fix some positive integer $m > 1$ and let $\mathfrak{D}$ be any domino system. Let $\mathrm{Sat}(\mathrm{SF})$ be the set containing all satisfiable SF sentences. We have $\mathrm{DOMINO}(\mathfrak{D}, 2^{\uparrow n}(m)) \leq_{n^2 \cdot \log n} \mathrm{Sat}(\mathrm{SF})$.

Having these reduction results at hand (we shall prove them shortly), Proposition 5.3.6 implies the following lower bounds regarding the time required to decide instances of $\mathrm{SF}_{\leq k}$-Sat and SF-Sat.

Theorem 5.3.10. *There are positive constants $c, d > 0$ for which*

$$\mathrm{Sat}(\mathrm{SF}_{\partial_\exists \leq k}) \notin \mathrm{NTIME}\big(2^{\uparrow k}(cn/\log n)\big)$$

and

$$\mathrm{Sat}(\mathrm{SF}) \notin \mathrm{NTIME}\big(2^{\uparrow d \cdot \sqrt{n/\log n}}(2)\big) \ .$$

Theorem 5.3.10 provides lower bounds regarding the time needed to decide SF-Sat and its subproblems in the worst-case. However, hardness for k-NExpTime, say, does not follow immediately. As Proposition 5.3.3 provides a reduction from the acceptance problem for nondeterministic $T(n)$-time-bounded Turing machines to bounded domino problems $\mathrm{DOMINO}(\mathfrak{D}, T(n))$ and Lemma 5.3.9 provides a reduction from such domino problems to subproblems of SF-Sat, we obtain the following hardness result.

Theorem 5.3.11. *For every positive integer k, the problem $\mathrm{SF}_{\leq k}$-Sat is k-NExpTime-hard.*

It is worth mentioning that Theorems 5.3.10 and 5.3.11 even hold for SF without equality, see Section 5.3.3. In the rest of the present section we outline the reductions described in Lemma 5.3.9.

5.3.1 Enforcing a Large Domain in SF

Recall that we intend to encode a given domino system $\mathfrak{D} = \langle \mathcal{D}, \mathcal{H}, \mathcal{V} \rangle$ — for nonempty $\mathcal{D}, \mathcal{H}, \mathcal{V}$ — plus an initial condition $\overline{D}$ into an SF sentence φ such that φ is satisfiable if and only if $\overline{D} \in \mathrm{DOMINO}(\mathfrak{D}, T_i(|\overline{D}|))$ with $T_1(n) = 2^{\uparrow\kappa}(n)$ for any fixed $\kappa > 0$ and $T_2(n) = 2^{\uparrow n}(\mu)$ for any fixed $\mu > 1$. The key issue in the encoding is the formalization of sufficiently large tori in SF. The following description gives a somewhat simplified view. Technical details will follow. For convenience, we allow the use of constant symbols in the encoding. This does certainly not change the computational complexity of SF-Sat.

We shall devise a satisfiable SF sentence whose models contain a grid of size $t \times t$, where t defines the required computing time and we assume $t := 2^{\uparrow\kappa}(\mu)$ for positive integers κ and $\mu > 1$ that we consider as parameters of the construction. Every point p on the grid is represented by a pair $p = \langle x, y \rangle$, where each of the coordinates x and y may assume a value from 0 to $2^{\uparrow\kappa}(\mu) - 1$. Each of the integers in that range is encoded by a bit string $\bar{b}$ of length $\log\big(2^{\uparrow\kappa}(\mu)\big) = 2^{\uparrow\kappa-1}(\mu)$.

κ, μ

The crux of our approach is that we have to enforce the existence of sufficiently many indices j, namely $2^{\uparrow\kappa-1}(\mu)$ many, to address the single bits of $\bar{b}$. Again, we address each of these indices using a bit string, this time of length $2^{\uparrow\kappa-2}(\mu)$. Thus, we proceed in an inductive fashion, building up a hierarchy of indices with $\kappa+1$ levels. The lowest level, level zero, is inhabited by μ indices, which we represent by constant symbols with pairwise distinct values. For every $\ell \geq 1$ any index j on the ℓ-th level is represented by a bit string consisting of $2^{\uparrow\ell-1}(\mu)$ bits, i.e. the ℓ-th level of the index hierarchy contains $2^{\uparrow\ell}(\mu)$ indices. The i-th bit of an ℓ-th-level index j corresponds to the truth value of the atom $J(\underline{\ell}, j, i)$, where $\underline{\ell}$ is a constant symbol used to address the ℓ-th level of the index hierarchy.

Example 5.3.12. *Assume $\mu = 2$ and $\kappa = 3$.*

index level	*set of indices*	*number of indices*
0	$\{\mathsf{c}_1, \mathsf{c}_2\}$	*2*
1	$\{00, 01, 10, 11\}$	*4*
2	$\{0000, 0001, \ldots, 1111\}$	*16*
3	$\{0,1\}^{16}$	65536

On every index level, the bits of one index are indexed by the indices from the previous level. We illustrate this for the word 1010 *on all levels from 2 down to 0. The bits of 1010 on level two are indexed by bit strings from level one, each of them having a length of two. The bits of the indices of level one are themselves indexed by objects of level zero which are some values $\mathsf{c}_1, \mathsf{c}_2$ assigned to the constant symbols c_1, c_2. To improve readability, we connect the bits of words by dashes.*

level 2: 1 — — — — 0 — — — — 1 — — — — 0
↑ ↑ ↑ ↑
level 1: 1 — 1 1 — 0 0 — 1 0 — 0
↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑
level 0: c_2 c_1 c_2 c_1 c_2 c_1 c_2 c_1

For technical reasons the number of indices per level shall grow slightly slower in our formalization than described above (cf. Lemma 5.3.14). The described index hierarchies can be encoded by SF formulas with the quantifier prefix $\exists^*(\forall\exists)^\kappa$ that have a length that is polynomial in κ and μ. We use the following constant and predicate symbols with the indicated meaning:

$\underline{0}, \underline{1}, \ldots, \underline{\kappa}$	constant symbols addressing the levels from 0 to κ,
$c_1, \ldots, c_\mu$	addresses the indices at level 0,
$d_1, \ldots, d_\kappa$	d_ℓ is the min. index at level ℓ,
$e_1, \ldots, e_\kappa$	e_ℓ is the max. index at level ℓ,
$L(\underline{\ell}, j)$	index j belongs to level ℓ,
$\mathrm{MinIdx}(\underline{\ell}, j)$	j is a min. index at level ℓ,
$\mathrm{MaxIdx}(\underline{\ell}, j)$	j is a max. index at level ℓ,
$J(\underline{\ell}, j, i, b)$	the i-th bit of the index j at level ℓ is b,
$J^*(\underline{\ell}, j, i, b)$	$b = 1$ indicates that all the bits of the index j that are less significant than j's i-th bit are 1,
$\mathrm{Succ}(\underline{\ell}, j, j')$	j' is the successor index of j at level ℓ.

On every level we implicitly establish an ordering over the indices of that level, from which we derive the concepts of minimal and maximal indices for each level. For this purpose, we use the usual ordering on natural numbers encoded in binary. Moreover, we formalize the usual successor relation by the predicate Succ. The idea underlying the formalization of binary successor is inspired by [BGG97] (proof of Theorem 6.2.13) and is sketched in the following example. It is based on the observation that a bit sting $b'_n \ldots b'_0$ is the successor of $b_n \ldots b_0$, if we have for every i, $0 \leq i \leq n$, that

$$b'_i = b_i \oplus (b_{i-1} \wedge \ldots \wedge b_0) \ ,$$

where $\oplus$ denotes the exclusive OR operation.

Example 5.3.13. *Consider the index $j := 1011$, which resides on the second level of the index hierarchy. The indices of the first level are* $00, 01, 10, 11$. *In terms of the predicates J and J^*, we*

get the following for the index j and its successors:

Representation of index j:

binary representation:	*1*	*0*	*1*	*1*
J-J^*-representation:	$J(\underline{2}, j, 11, 1)$	$J(\underline{2}, j, 10, 0)$	$J(\underline{2}, j, 01, 1)$	$J(\underline{2}, j, 00, 1)$
	$J^*(\underline{2}, j, 11, 0)$	$J^*(\underline{2}, j, 10, 1)$	$J^*(\underline{2}, j, 01, 1)$	$J^*(\underline{2}, j, 00, 1)$

Representation of the successor index j':

binary representation:	*1*	*1*	*0*	*0*
J-J^*-representation:	$J(\underline{2}, j', 11, 1)$	$J(\underline{2}, j', 10, 1)$	$J(\underline{2}, j', 01, 0)$	$J(\underline{2}, j', 00, 0)$
	$J^*(\underline{2}, j', 11, 0)$	$J^*(\underline{2}, j', 10, 0)$	$J^*(\underline{2}, j', 01, 0)$	$J^*(\underline{2}, j', 00, 1)$

Representation of the second successor index j'':

binary representation:	*1*	*1*	*0*	*1*
J-J^*-representation:	$J(\underline{2}, j'', 11, 1)$	$J(\underline{2}, j'', 10, 1)$	$J(\underline{2}, j'', 01, 0)$	$J(\underline{2}, j'', 00, 1)$
	$J^*(\underline{2}, j'', 11, 0)$	$J^*(\underline{2}, j'', 10, 0)$	$J^*(\underline{2}, j'', 01, 1)$	$J^*(\underline{2}, j'', 00, 1)$

What we observe is that the J-representation of j's successor j' is the result of combining j's J-representation with its J^*-representation using bitwise exclusive OR. More technically, for every $i \in \{00, 01, 10, 11\}$ and all $b, b^* \in \{0, 1\}$ we have that $J(\underline{2}, j, i, b)$ and $J^*(\underline{2}, j, i, b_*)$ entails $J(\underline{2}, j', i, b \oplus b_*)$.

A major difficulty that we encounter is that we cannot assert the existence of successors simply by adding $\forall j \exists j'.\, \mathrm{Succ}(\underline{\ell}, j, j')$, as j and j' would not be separated. Therefore, we fall back on a trick that we have already used in Section 3.3.3 (cf. pages 42–45) when we investigated the expressiveness of SF when only models of a bounded size are considered. We start from the equivalent formula $\forall j \exists \widetilde{j} j'.\, j \approx \widetilde{j} \wedge \mathrm{Succ}(\underline{\ell}, \widetilde{j}, j')$, and replace the atom $j \approx \widetilde{j}$ by a subformula $\mathrm{eq}^{\ell}_{j,\widetilde{j}}$ in which j and $\widetilde{j}$ are separated and which expresses a certain similarity between j and $\widetilde{j}$ instead of identity. However, we specify the hierarchy of indices in a sufficiently strong way such that the similarity expressed by $\mathrm{eq}^{\ell}_{j,\widetilde{j}}$ actually coincides with identity. The subformula $\mathrm{eq}^{\ell}_{j,\widetilde{j}}$ will also reoccur in other places, namely, whenever we need to enforce the existence of successors of one kind or another. It will be the main source of nested quantifier alternations. The depth of the nesting depends linearly on the parameter ℓ, which will become at most κ in the construction.

Next, we formalize the described index hierarchies in $\mathrm{SF}_{\leq\kappa}$. Technically speaking, we will be more liberal than the formal SF syntax allows in that the given sentences will contain constant symbols and will not be in prenex normal form. However, equivalent SF sentences of (almost) the same length can be derived easily. Every formula is accompanied by a brief description of its purpose. We shall try to use as few essentially non-Horn sentences as possible.

$$\psi_1 := \bigwedge_{\ell=0}^{\kappa} \bigwedge_{\substack{\ell'=0 \\ \ell' \neq \ell}}^{\kappa} \forall j.\, L(\underline{\ell}, j) \rightarrow \neg L(\underline{\ell'}, j)$$

Every index belongs to at most one level.

$$\psi_2 := \bigwedge_{\ell=0}^{\kappa} \big(\forall j.\, \mathrm{MinIdx}(\underline{\ell}, j) \rightarrow L(\underline{\ell}, j)\big) \;\wedge\; \big(\forall j j'.\, \mathrm{MinIdx}(\underline{\ell}, j) \rightarrow \neg \mathrm{Succ}(\underline{\ell}, j', j)\big)$$

A min. index of level ℓ belongs to level ℓ. A min. index does not have a predecessor.

$$\psi_3 := \bigwedge_{\ell=0}^{\kappa} \mathrm{MinIdx}(\underline{\ell}, d_\ell) \;\wedge\; \big(\forall j.\, \mathrm{MinIdx}(\underline{\ell}, j) \rightarrow j \approx d_\ell\big)$$

There is a unique min. index on every level.

$$\psi_4 := \bigwedge_{\ell=0}^{\kappa} \big(\forall j.\, \mathrm{MaxIdx}(\underline{\ell}, j) \rightarrow L(\underline{\ell}, j)\big) \;\wedge\; \big(\forall j j'.\, \mathrm{MaxIdx}(\underline{\ell}, j) \rightarrow \neg \mathrm{Succ}(\underline{\ell}, j, j')\big)$$

A max. index of level ℓ belongs to level ℓ. A max. index does not have a successor.

$$\psi_5 := \bigwedge_{\ell=0}^{\kappa} \mathrm{MaxIdx}(\underline{\ell}, e_\ell) \quad \wedge \quad \big(\forall j.\ \mathrm{MaxIdx}(\underline{\ell}, j) \to j \approx e_\ell\big)$$

There is a unique max. index on every level.

$$\psi_6 := \bigwedge_{\ell=0}^{\kappa} \forall jj'.\ \mathrm{Succ}(\underline{\ell}, j, j') \quad \to \quad L(\underline{\ell}, j) \wedge L(\underline{\ell}, j')$$

If j' is the successor of j at level ℓ, then both j and j' belong to level ℓ.

$$\psi_7 := \bigwedge_{\ell=0}^{\kappa} \forall jj'j''.\ \neg\mathrm{Succ}(\underline{\ell}, j, j) \quad \wedge \quad \big(\mathrm{Succ}(\underline{\ell}, j, j') \wedge \mathrm{Succ}(\underline{\ell}, j, j'') \to j' \approx j''\big)$$
$$\wedge \quad \big(\mathrm{Succ}(\underline{\ell}, j', j) \wedge \mathrm{Succ}(\underline{\ell}, j'', j) \to j' \approx j''\big)$$

The successor relation is irreflexive. Every index j has at most one successor and at most one predecessor.

$$\psi_8 := \mathrm{MinIdx}(\underline{0}, c_1) \wedge \mathrm{MaxIdx}(\underline{0}, c_\mu) \quad \wedge \quad \bigwedge_{i=1}^{\mu-1} \mathrm{Succ}(\underline{0}, c_i, c_{i+1})$$

At level zero we have the sequence $c_1, \ldots, c_\mu$ of successors, where c_1 is min. and c_μ max.

$$\psi_9 := \bigwedge_{\ell=1}^{\kappa} \forall jj'i.\ \mathrm{Succ}(\underline{\ell}, j, j') \wedge L(\underline{\ell-1}, i) \quad \to \quad \Big(\big(J^*(\underline{\ell}, j, i, 1) \wedge J(\underline{\ell}, j, i, 1) \to J(\underline{\ell}, j', i, 0)\big)$$
$$\wedge \big(J^*(\underline{\ell}, j, i, 1) \wedge J(\underline{\ell}, j, i, 0) \to J(\underline{\ell}, j', i, 1)\big)$$
$$\wedge \big(J^*(\underline{\ell}, j, i, 0) \wedge J(\underline{\ell}, j, i, 1) \to J(\underline{\ell}, j', i, 1)\big)$$
$$\wedge \big(J^*(\underline{\ell}, j, i, 0) \wedge J(\underline{\ell}, j, i, 0) \to J(\underline{\ell}, j', i, 0)\big)\Big)$$

Define what it means to be a successor at level ℓ, $\ell > 0$, in terms of the binary increment operation modulo $2^{\uparrow\ell}(\mu)$. This formula resembles the bitwise exclusive OR operation as illustrated in Example 5.3.13.

$$\psi_{10} := \bigwedge_{\ell=1}^{\kappa} \forall ji.\ \mathrm{MinIdx}(\underline{\ell}, j) \wedge L(\underline{\ell-1}, i) \quad \to \quad J(\underline{\ell}, j, i, 0)$$

All bits of a minimal index j are 0.

$$\psi_{11} := \bigwedge_{\ell=1}^{\kappa} \forall ji.\ \mathrm{MaxIdx}(\underline{\ell}, j) \wedge \mathrm{MaxIdx}(\underline{\ell-1}, i) \quad \to \quad J(\underline{\ell}, j, i, 1)$$

Define what it means to be max. (part 1): the most significant bit is 1.

$$\psi_{12} := \bigwedge_{\ell=1}^{\kappa} \forall ji.\ L(\underline{\ell}, j) \wedge \mathrm{MaxIdx}(\underline{\ell-1}, i) \wedge J(\underline{\ell}, j, i, 1) \quad \to \quad \mathrm{MaxIdx}(\underline{\ell}, j)$$

Define what it means to be max. (part 2): any index with 1 as its most significant bit is max.

$$\psi_{13} := \bigwedge_{\ell=1}^{\kappa} \forall ji.\ L(\underline{\ell}, j) \wedge L(\underline{\ell-1}, i) \quad \to \quad \big(J(\underline{\ell}, j, i, 0) \to \neg J(\underline{\ell}, j, i, 1)\big)$$
$$\wedge \big(J^*(\underline{\ell}, j, i, 0) \to \neg J^*(\underline{\ell}, j, i, 1)\big)$$

No bit of an index is 0 and 1 at the same time. An analogous requirement is stipulated for J^*.

$$\psi_{14} := \bigwedge_{\ell=1}^{\kappa} \forall ji.\ L(\underline{\ell}, j) \wedge \mathrm{MinIdx}(\underline{\ell-1}, i) \quad \to \quad J^*(\underline{\ell}, j, i, 1)$$

We stipulate $J^*(\underline{\ell}, j, d_{\ell-1}, 1)$ for every index j.

$$\psi_{15} := \bigwedge_{\ell=1}^{\kappa} \forall jii'.\, L(\underline{\ell}, j) \wedge \mathrm{Succ}(\underline{\ell-1}, i, i') \;\rightarrow\; \big(J^*(\underline{\ell}, j, i', 1) \leftrightarrow \big(J^*(\underline{\ell}, j, i, 1) \wedge J(\underline{\ell}, j, i, 1)\big)\big) \wedge \big(J(\underline{\ell}, j, i, 0) \rightarrow J^*(\underline{\ell}, j, i', 0)\big) \wedge \big(J^*(\underline{\ell}, j, i, 0) \rightarrow J^*(\underline{\ell}, j, i', 0)\big)$$

Define the semantics of J^* as indicating that all bits strictly less significant than the i-th bit are 1.

$$\mathrm{eq}^1_{j,\widetilde{j}} := L(\underline{1}, j) \wedge L(\underline{1}, \widetilde{j}) \;\wedge\; \bigwedge_{i=1}^{\mu} \big(J(\underline{1}, j, c_i, 0) \leftrightarrow J(\underline{1}, \widetilde{j}, c_i, 0)\big) \wedge \big(J(\underline{1}, j, c_i, 1) \leftrightarrow J(\underline{1}, \widetilde{j}, c_i, 1)\big)$$

Base case of equality of indices.

$$\mathrm{eq}^{\ell}_{j,\widetilde{j}} := L(\underline{\ell}, j) \wedge L(\underline{\ell}, \widetilde{j}) \wedge \forall i.\, L(\underline{\ell-1}, i) \;\rightarrow\; \exists\widetilde{i}.\, L(\underline{\ell-1}, \widetilde{i}) \wedge \mathrm{eq}^{\ell-1}_{i,\widetilde{i}} \wedge \big(J(\underline{\ell}, j, i, 0) \leftrightarrow J(\underline{\ell}, \widetilde{j}, \widetilde{i}, 0)\big) \wedge \big(J(\underline{\ell}, j, i, 1) \leftrightarrow J(\underline{\ell}, \widetilde{j}, \widetilde{i}, 1)\big)$$

Inductive case of equality of indices for $\ell > 1$.

$$\psi_{16} := \bigwedge_{\ell=1}^{\kappa} \forall ji.\, L(\underline{\ell}, j) \wedge \mathrm{MaxIdx}(\underline{\ell-1}, i) \wedge J(\underline{\ell}, j, i, 0) \;\rightarrow\; \exists\widetilde{j}.\, \mathrm{eq}^{\ell}_{j,\widetilde{j}} \wedge \exists\widetilde{j}'.\, \mathrm{Succ}(\underline{\ell}, \widetilde{j}, \widetilde{j}')$$

For every index at level ℓ that is not maximal, i.e. whose most significant bit is 0, there exists a successor index.

Until now, we have only introduced sentences that can easily be transformed into SF sentences in Horn form. This follows from the fact that all consequents of implications are either literals or conjunctions of literals. Moreover, all existentially quantified variables are separated from universally quantified variables. To verify this, simple inspection of the formulas suffices, since all quantifiers occur with positive polarity, i.e. within the scope of an even number of (implicit) negation signs.

Regarding the length of the above sentences, we observe the following:

$$\mathrm{len}(\psi_1) \in \mathcal{O}(\kappa^2 \log \kappa),$$

$$\mathrm{len}(\psi_2), \ldots, \mathrm{len}(\psi_7), \mathrm{len}(\psi_9), \ldots, \mathrm{len}(\psi_{15}) \in \mathcal{O}(\kappa \log \kappa),$$

$$\mathrm{len}(\psi_8) \in \mathcal{O}\big(\mu(\log \kappa + \log \mu)\big),$$

$$\mathrm{len}\big(\mathrm{eq}^1_{j,j'}\big) \in \mathcal{O}\big(\mu(\log \kappa + \log \mu)\big),$$

$$\mathrm{len}\big(\mathrm{eq}^{\ell}_{j,j'}\big) \in \mathcal{O}(\log \kappa) + \mathrm{len}\big(\mathrm{eq}^{\ell-1}_{j,j'}\big),$$

$$\mathrm{len}(\psi_{16}) \in \mathcal{O}\big(\kappa^2 \log \kappa + \kappa\mu(\log \kappa + \log \mu)\big).$$

In total, this yields $\mathrm{len}(\psi_1 \wedge \ldots \wedge \psi_{16}) \in \mathcal{O}\big(\kappa^2 \log \kappa + \kappa\mu(\log \kappa + \log \mu)\big)$.

The following three sentences do not produce Horn formulas when transformed into CNF. They serve the purpose of removing spurious elements from the model. In particular, χ_3 is essential to enforce large models for $\kappa \geq 2$.

$$\chi_1 := \forall j.\, L(\underline{0}, j) \rightarrow \bigvee_{i=1}^{\mu} j \approx c_i$$

On level 0 there are no indices but $c_1, \ldots, c_\mu$.

$$\chi_2 := \bigwedge_{\ell=1}^{\kappa} \forall ji.\, L(\underline{\ell}, j) \wedge L(\underline{\ell-1}, i) \;\rightarrow\; J(\underline{\ell}, j, i, 0) \vee J(\underline{\ell}, j, i, 1)$$

We stipulate totality for the predicate J.

$$\chi_3 := \bigwedge_{\ell=1}^{\kappa} \forall jj'.\, L(\underline{\ell}, j) \wedge L(\underline{\ell}, j') \to \exists\widetilde{j}.\, \mathrm{eq}^{\ell}_{j,\widetilde{j}} \wedge \exists\widetilde{j}'.\, \mathrm{eq}^{\ell}_{j',\widetilde{j}'} \\ \wedge \Big(\Big(\forall\widetilde{i}.\, L(\underline{\ell-1}, \widetilde{i}) \to \big(J(\underline{\ell}, \widetilde{j}, \widetilde{i}, 0) \leftrightarrow J(\underline{\ell}, \widetilde{j}', \widetilde{i}, 0)\big)\Big) \to j \approx j'\Big)$$

Two indices at the same level that agree on all of their bits are required to be identical.

Notice that χ_3 is (almost) an SF sentence, since the $\forall\widetilde{i}$ turns into a $\exists\widetilde{i}$ as soon as we bring the sentence into prenex normal form. Regarding the length of χ_1, χ_2, χ_3, we observe $\mathrm{len}(\chi_1) \in \mathcal{O}(\log\kappa + \mu\log\mu)$, $\mathrm{len}(\chi_2) \in \mathcal{O}(\kappa\log\kappa)$, and $\mathrm{len}(\chi_3) \in \mathcal{O}\big(\kappa^2\log\kappa + \kappa\mu(\log\kappa + \log\mu)\big)$. Hence, we overall have $\mathrm{len}(\chi_1 \wedge \chi_2 \wedge \chi_3) \in \mathcal{O}\big(\kappa^2\log\kappa + \kappa\mu(\log\kappa + \log\mu)\big)$.

This finishes the formalization of the index hierarchy. Before we go on with the formalization of tiling problems — starting from page 163 —, we shall make sure that the formalization of the hierarchy is correct. More concretely, we aim to show that in any model $\mathcal{A}$ of the sentence $\psi_1 \wedge \ldots \wedge \psi_{16} \wedge \chi_1 \wedge \chi_2 \wedge \chi_3$ every level ℓ contains exactly $2^{\uparrow\ell}(\mu-1)+1$ indices and that for every level ℓ the successor relation Succ induces a unique chain in which all the indices of level ℓ are lined up in a linear fashion. To this end, for the rest of this subsection, we fix any model $\mathcal{A}$ of the sentence $\psi_1 \wedge \ldots \wedge \psi_{16} \wedge \chi_1 \wedge \chi_2 \wedge \chi_3$.

Lemma 5.3.14 (Short version of Lemma 5.3.23)**.** *For every $\ell = 0, \ldots, \kappa$ let* $\mathcal{I}_\ell$

$$\mathcal{I}_\ell := \big\{\mathsf{a} \in \mathsf{A} \mid \mathcal{A} \models L(\underline{\ell}, \mathsf{a})\big\}$$

and let the relation $\prec_\ell \subseteq \mathcal{I}_\ell \times \mathcal{I}_\ell$ be defined such that $\mathsf{a} \prec_\ell \mathsf{a}'$

$$\mathsf{a} \prec_\ell \mathsf{a}' \text{ holds if and only if } \mathcal{A} \models \mathrm{Succ}(\underline{\ell}, \mathsf{a}, \mathsf{a}').$$

Then, for every $\ell = 1, \ldots, \kappa$ we have $|\mathcal{I}_\ell| = p$ where $p := 2^{|\mathcal{I}_{\ell-1}|-1} + 1 = 2^{\uparrow\ell}(\mu-1)+1$. Moreover, there is a unique chain $\mathsf{a}_1 \prec_\ell \ldots \prec_\ell \mathsf{a}_p$ comprising all elements in $\mathcal{I}_\ell$, and we have $\mathsf{a}_1 = d^{\mathcal{A}}_\ell$ and $\mathsf{a}_p = e^{\mathcal{A}}_\ell$.

Leaving out the non-Horn parts χ_1, χ_2, χ_3 renders the lemma invalid for $\ell > 1$. On the other hand, for $\kappa = 1$ the sentence $\psi_1 \wedge \ldots \wedge \psi_{16}$ — which can be transformed into an equivalent Horn sentence — has only models $\mathcal{A}$ for which $\mathcal{I}_1$ contains at least $2^{\mu-1}+1$ elements. This observation could be used to derive ExpTime-hardness of Horn-SF$_{\leq 1}$-Sat. But since the subproblems Horn-MFO-Sat and Horn-BSR-Sat are already known to be ExpTime-hard (cf. Table 5.1), we would not gain new insights.

We now embark on proving Lemma 5.3.14 (via proving its extended version, Lemma 5.3.23).

Definition 5.3.15. *In addition to the sets and relations defined in Lemma 5.3.14, we fix the following notation:*

For every $\ell = 0, \ldots, \kappa-1$ we define

$\mathcal{F}_\ell$ $\quad \mathcal{F}_\ell := \{f : \mathcal{I}_\ell \to \{0,1\} \mid f$ *is total and for every* a *with* $\mathcal{A} \models \mathrm{MaxIdx}(\underline{\ell}, \mathsf{a})$ *we have that* $f(\mathsf{a}) = 1$ *entails* $f(\mathsf{b}) = 0$ *for every* $\mathsf{b} \neq \mathsf{a}\}$.

(Intended meaning: Suppose, there is a unique chain $\mathsf{a}_1 \prec_\ell \ldots \prec_\ell \mathsf{a}_n$ comprising all elements from $\mathcal{I}_\ell$. Then, every function f in $\mathcal{F}_\ell$ corresponds to a bit string $f(\mathsf{a}_1)\ldots f(\mathsf{a}_n)$ (with the least significant bit on the left). Unless we can prove the uniqueness of the above chain, f merely remains an unordered collection of bits, each addressed by some elements from $\mathcal{I}_\ell$. The special requirements towards the functions in $\mathcal{F}_\ell$ regarding maximal indices is due to the technical detail that we define maximal indices to be the ones that are represented by bit strings of the form $10\ldots0$.)

For every $\ell = 0, \ldots, \kappa-1$ and any two total functions $f, g : \mathcal{I}_\ell \to \{0,1\}$ we write $f \sqsubset_\ell g$ if and only if (a) there is some integer p and a unique chain $\mathsf{a}_1 \prec_\ell \ldots \prec_\ell \mathsf{a}_p$ comprising all elements in $\mathcal{I}_\ell$, and (b) incrementing the bit string $f(\mathsf{a}_1)f(\mathsf{a}_2)\ldots f(\mathsf{a}_p)$ (interpreted as number encoded in binary where the leftmost bit is the least significant one) by one yields $g(\mathsf{a}_1)g(\mathsf{a}_2)\ldots g(\mathsf{a}_p)$. $f \sqsubset_\ell g$

For every $\ell = 1, \ldots, \kappa$ and every $f \in \mathcal{F}_{\ell-1}$ we define

$\mathcal{S}_{\ell,f}$

$$\mathcal{S}_{\ell,f} := \left\{ \mathsf{a} \in \mathcal{I}_\ell \mid \mathcal{A} \models J(\underline{\ell}, \mathsf{a}, \mathsf{b}, f(\mathsf{b})) \text{ for every } \mathsf{b} \in \mathcal{I}_{\ell-1} \right\}.$$

(Intended meaning: Suppose, there is a unique chain $\mathsf{b}_1 \prec_{\ell-1} \ldots \prec_{\ell-1} \mathsf{b}_n$ comprising all elements from $\mathcal{I}_{\ell-1}$. Then, the set $S_{\ell,f} \subseteq \mathcal{I}_\ell$ comprises all elements from $\mathcal{I}_\ell$ that are represented by the bit string $f(\mathsf{b}_1) \ldots f(\mathsf{b}_n)$. We aim to prove that every $S_{\ell,f}$ contains exactly one element or, in other words, that every $f \in \mathcal{F}_{\ell-1}$ represents exactly one element from $\mathcal{I}_\ell$.)

$\mathsf{a} \sim_\ell \mathsf{a}'$

For every $\ell = 1, \ldots, \kappa$ and any two elements $\mathsf{a}, \mathsf{a}' \in \mathcal{I}_\ell$ we write $\mathsf{a} \sim_\ell \mathsf{a}'$ if and only if for every $\mathsf{b} \in \mathcal{I}_{\ell-1}$ we observe

$\mathcal{A} \models J(\underline{\ell}, \mathsf{a}, \mathsf{b}, 0)$ *if and only if* $\mathcal{A} \models J(\underline{\ell}, \mathsf{a}', \mathsf{b}, 0)$ *and*

$\mathcal{A} \models J(\underline{\ell}, \mathsf{a}, \mathsf{b}, 1)$ *if and only if* $\mathcal{A} \models J(\underline{\ell}, \mathsf{a}', \mathsf{b}, 1)$.

Proposition 5.3.16. *For every ℓ, $1 \leq \ell \leq \kappa$, $\sim_\ell$ is an equivalence relation.*

Proposition 5.3.17. *For all distinct ℓ, ℓ', $0 \leq \ell, \ell' \leq \kappa$, $\mathcal{I}_\ell$ and $\mathcal{I}_{\ell'}$ are disjoint.*

Proof. This is a direct consequence of $\mathcal{A} \models \psi_1$. □

Lemmas 5.3.18 to 5.3.22 are auxiliary technical lemmas that will be used in the proof of Lemma 5.3.23. Recall that the relation $\prec_\ell$ over $\mathcal{I}_\ell$ represents the successor relation on index level ℓ induced by the predicate $\mathrm{Succ}^{\mathcal{A}}$. Further recall that the relation $\sqsubset_{\ell-1}$ over $\mathcal{F}_{\ell-1}$ is intended to represent the successor relation over bit strings of length $|\mathcal{I}_{\ell-1}|$. Roughly speaking, Lemma 5.3.18 shows that, if we link the functions in $\mathcal{F}_{\ell-1}$ with the elements in $\mathcal{I}_\ell$ via the predicate $J^{\mathcal{A}}$, then the relation $\sqsubset_{\ell-1}$ reflects the relation $\prec_\ell$. Lemmas 5.3.19 and 5.3.20 together entail that the sets $S_{\ell,f} \subseteq \mathcal{I}_\ell$ resemble the equivalence classes induced by $\sim_\ell$. Lemma 5.3.21 states that the functions in any $\mathcal{F}_\ell$ can be uniquely arranged into a chain $f_1 \sqsubset_\ell \ldots \sqsubset_\ell f_{p'}$, provided that there is a corresponding unique chain $\mathsf{a}_1 \prec_\ell \ldots \prec_\ell \mathsf{a}_p$ of all elements from $\mathcal{I}_\ell$. Lemma 5.3.22 establishes the existence of such a unique chain $\mathsf{a}_1 \prec_0 \ldots \prec_0 \mathsf{a}_\mu$ for the elements in $\mathcal{I}_0$. Finally, Lemma 5.3.23 brings it all together and, roughly speaking, states that (a) the relations $\sim_\ell$ coincide with the semantics of $\mathrm{eq}^\ell_{j,\tilde{j}}$ under $\mathcal{A}$, (b) each of the sets $S_{\ell,f}$, which resemble the equivalence classes of $\sim_\ell$, contains exactly one element, and (c) the elements of any set $\mathcal{I}_\ell$ can be uniquely arranged in a chain $\mathsf{a}_1 \prec_\ell \ldots \prec_\ell \mathsf{a}_p$ with $p = |\mathcal{F}_{\ell-1}| = 2^{\uparrow\ell}(\mu - 1) + 1$.

Lemma 5.3.18. *Let $\mathsf{a}, \mathsf{a}' \in \mathcal{I}_\ell$ for some ℓ, $1 \leq \ell \leq \kappa$. Let $f, g : \mathcal{I}_{\ell-1} \to \{0,1\}$ be two total functions such that for every $\mathsf{b} \in \mathcal{I}_{\ell-1}$ we have $\mathcal{A} \models J(\underline{\ell}, \mathsf{a}, \mathsf{b}, f(\mathsf{b}))$ and $\mathcal{A} \models J(\underline{\ell}, \mathsf{a}', \mathsf{b}, g(\mathsf{b}))$. Moreover, we assume that there is a unique chain $\mathsf{c}_1 \prec_{\ell-1} \ldots \prec_{\ell-1} \mathsf{c}_p$ comprising all elements in $\mathcal{I}_{\ell-1}$. Then $\mathsf{a} \prec_\ell \mathsf{a}'$ implies $f \sqsubset_{\ell-1} g$.*

Proof. Since c_1 is the only element in $\mathcal{I}_{\ell-1}$ for which there is no element $\mathsf{c}' \in \mathcal{I}_{\ell-1}$ with $\mathsf{c}' \prec_{\ell-1} \mathsf{c}_1$, $\mathcal{A} \models \psi_2 \wedge \psi_3$ implies $\mathcal{A} \models \mathrm{MinIdx}(\underline{\ell - 1}, \mathsf{c}_1)$.

Let $f^* : \mathcal{I}_{\ell-1} \to \{0,1\}$ be defined such that

$f^*(\mathsf{b}) = 0$ if and only if $\mathcal{A} \models J^*(\underline{\ell}, \mathsf{a}, \mathsf{b}, 0)$ and

$f^*(\mathsf{b}) = 1$ if and only if $\mathcal{A} \models J^*(\underline{\ell}, \mathsf{a}, \mathsf{b}, 1)$.

This function is well-defined because of $\mathcal{A} \models \psi_{13}$. Due to $\mathcal{A} \models \chi_2 \wedge \psi_{14} \wedge \psi_{15}$ it is also total. $\mathcal{A} \models \psi_{14}$ enforces $\mathcal{A} \models J^*(\underline{\ell}, \mathsf{a}, \mathsf{c}_1, 1)$, i.e. $f^*(\mathsf{c}_1) = 1$. Moreover, for any k with $1 < k \leq p$ we have $f^*(\mathsf{c}_k) = 1$ if and only if $f(\mathsf{c}_1) = \ldots = f(\mathsf{c}_{k-1}) = 1$, because of $\mathcal{A} \models \psi_{15}$. $\mathcal{A} \models \psi_9$ together with our assumption $\mathsf{a} \prec_\ell \mathsf{a}'$ translates to the following property, which we phrase in terms of operations on bits: for every k, $1 \leq k \leq p$, we observe $g(\mathsf{c}_k) = f(\mathsf{c}_k) \oplus f^*(\mathsf{c}_k)$ where $\oplus$ denotes exclusive OR. But this corresponds to an increment of the bit string $f(\mathsf{c}_1) \ldots f(\mathsf{c}_p)$ by one (where $f(\mathsf{c}_1)$ is the least significant bit). Hence, $f \sqsubset_{\ell-1} g$. □

Lemma 5.3.19. *Let $\mathsf{a}, \mathsf{a}' \in \mathcal{I}_\ell$ for some ℓ, $1 \leq \ell \leq \kappa$. If a and a' belong to the same $\mathcal{S}_{\ell,f}$ for some function $f \in \mathcal{F}_{\ell-1}$, then $\mathsf{a} \sim_\ell \mathsf{a}'$.*

Proof. By totality of f, it follows that for every $\mathsf{b} \in \mathcal{I}_{\ell-1}$ we have $\mathcal{A} \models J(\underline{\ell}, \mathsf{a}, \mathsf{b}, f(\mathsf{b}))$ and $\mathcal{A} \models J(\underline{\ell}, \mathsf{a}', \mathsf{b}, f(\mathsf{b}))$. Moreover, $\mathcal{A} \models \psi_{13}$ entails $\mathcal{A} \not\models J(\underline{\ell}, \mathsf{a}, \mathsf{b}, 1-f(\mathsf{b}))$ and $\mathcal{A} \not\models J(\underline{\ell}, \mathsf{a}', \mathsf{b}, 1-f(\mathsf{b}))$. This results in $\mathsf{a} \sim_\ell \mathsf{a}'$. □

Lemma 5.3.20. *Let $\mathsf{a}, \mathsf{a}' \in \mathcal{I}_\ell$ for some ℓ, $1 \leq \ell \leq \kappa$. Moreover, let f be some function in $\mathcal{F}_{\ell-1}$. If a belongs to $\mathcal{S}_{\ell,f}$ and we have $\mathsf{a} \sim_\ell \mathsf{a}'$, then $\mathsf{a}' \in \mathcal{S}_{\ell,f}$.*

Proof. By totality of f, it follows that for every $\mathsf{b} \in \mathcal{I}_{\ell-1}$ we have $\mathcal{A} \models J(\underline{\ell}, \mathsf{a}, \mathsf{b}, f(\mathsf{b}))$. Moreover, $\mathcal{A} \models \psi_{13}$ entails $\mathcal{A} \not\models J(\underline{\ell}, \mathsf{a}, \mathsf{b}, 1 - f(\mathsf{b}))$. Since we assume $\mathsf{a} \sim_\ell \mathsf{a}'$, we know that these properties transfer to a'. Hence, for every $\mathsf{b} \in \mathcal{I}_{\ell-1}$ we observe $\mathcal{A} \models J(\underline{\ell}, \mathsf{a}', \mathsf{b}, f(\mathsf{b}))$ and $\mathcal{A} \not\models J(\underline{\ell}, \mathsf{a}', \mathsf{b}, 1-f(\mathsf{b}))$. Consequently, $\mathsf{a}' \in \mathcal{S}_{\ell,f}$. □

Lemma 5.3.21. *Consider any ℓ, $0 \leq \ell \leq \kappa - 1$. If there is a unique chain $\mathsf{a}_1 \prec_\ell \ldots \prec_\ell \mathsf{a}_p$ comprising all elements in $\mathcal{I}_\ell$, then we can uniquely arrange all functions in $\mathcal{F}_\ell$ into a chain $f_1 \sqsubset_\ell \ldots \sqsubset_\ell f_{p'}$ where $p' = |\mathcal{F}_\ell| = 2^{p-1} + 1$.*

Proof. Let $\{0,1\}^p$ be the set of all bit strings of length p. If we interpret each of them as a number encoded in binary (where we assume the rightmost bit to be the least significant one), we can uniquely arrange the $2^{p-1} + 1$ smallest bit strings in $\{0,1\}^p$ into a chain

$$\bar{b}_0 < \bar{b}_1 < \ldots < \bar{b}_{2^{p-1}-1} < \bar{b}_{2^{p-1}}$$

where the indices reflect the encoded numerical value and $<$ is intended to be the usual ordering based on this value. Since we assume the leftmost bit to be the most significant one, it is 0 in $\bar{b}_0, \ldots, \bar{b}_{2^{p-1}-1}$. Accordingly, $\bar{b}_{2^{p-1}}$ is the bit string with all zeros except for the most significant bit, i.e. $\bar{b}_{2^{p-1}} = 10\ldots0$.

Obviously, the following mapping ρ induces a one-to-one correspondence between bit strings and all the mappings in $\mathcal{F}_\ell$: $\rho(f) := f(\mathsf{a}_p)f(\mathsf{a}_{p-1})\ldots f(\mathsf{a}_2)f(\mathsf{a}_1)$. By definition of $\sqsubset_\ell$, we have $f \sqsubset_\ell g$ if and only if $\rho(f) + 1 = \rho(g)$. Consequently, we obtain the chain $\rho^{-1}(\bar{b}_0) \sqsubset_\ell \rho^{-1}(\bar{b}_1) \sqsubset_\ell \ldots \sqsubset_\ell \rho^{-1}(\bar{b}_{2^{p-1}-1}) \sqsubset_\ell \rho^{-1}(\bar{b}_{2^{p-1}})$. □ ρ

Lemma 5.3.22. *$\mathcal{I}_0$ contains exactly the elements $c_1^\mathcal{A}, \ldots, c_\mu^\mathcal{A}$, and these are pairwise distinct. Moreover, there is a unique chain $c_1^\mathcal{A} \prec_0 \ldots \prec_0 c_\mu^\mathcal{A}$.*

Proof. $\mathcal{A} \models \chi_1$ entails that $\mathcal{I}_0 \subseteq \{c_1^\mathcal{A}, \ldots, c_\mu^\mathcal{A}\}$. Due to $\mathcal{A} \models \psi_8$ we have $c_1^\mathcal{A} \prec_0 \ldots \prec_0 c_\mu^\mathcal{A}$. By $\mathcal{A} \models \psi_6$, $\mathsf{a} \prec_0 \mathsf{b}$ entails $\mathsf{a}, \mathsf{b} \in \mathcal{I}_0$. Hence, $\{c_1^\mathcal{A}, \ldots, c_\mu^\mathcal{A}\} \subseteq \mathcal{I}_0$.

We next show that all $c_1^\mathcal{A}, \ldots, c_\mu^\mathcal{A}$ are pairwise distinct.

Claim: For every index $j \geq 2$ the first j elements $c_1^\mathcal{A}, \ldots, c_j^\mathcal{A}$ are distinct.

Proof: We proceed by induction on j.

For $j = 2$, $c_1^\mathcal{A} \neq c_2^\mathcal{A}$ must hold, for otherwise $c_1^\mathcal{A} \prec_0 c_2^\mathcal{A}$ contradicts $\mathcal{A} \models \psi_7$ which entails $\mathcal{A} \models \forall j.\, \neg\mathrm{Succ}(\underline{0}, j, j)$.

Let $j \geq 3$ and assume, by induction, that the elements $c_1^\mathcal{A}, \ldots, c_{j-1}^\mathcal{A}$ are all pairwise distinct. Suppose there is some index i_*, $1 \leq i_* < j$, such that $c_{i_*}^\mathcal{A} = c_j^\mathcal{A}$. We distinguish two cases: In case $i_* = 1$, we have $c_{j-1}^\mathcal{A} \prec_0 c_1^\mathcal{A}$. But this contradicts $\mathcal{A} \models \psi_2$. In case of $i_* > 1$, $\mathcal{A} \models \psi_7$ entails $c_{i_*-1}^\mathcal{A} = c_{j-1}^\mathcal{A}$, since we have $c_{i_*-1}^\mathcal{A} \prec_0 c_{i_*}^\mathcal{A}$ and $c_{j-1}^\mathcal{A} \prec_0 c_j^\mathcal{A}$, and since we assumed $c_{i_*}^\mathcal{A} = c_j^\mathcal{A}$. But this contradicts our inductive hypothesis, because $i_* - 1$ and $j - 1$ are distinct indices and thus the inductive hypothesis implies that $c_{i_*-1}^\mathcal{A}$ and $c_{j-1}^\mathcal{A}$ are distinct. ◊

The above claim entails $|\mathcal{I}_0| = \mu$.

$\mathcal{A} \models \psi_8$ entails $c_1^\mathcal{A} \prec_0 \ldots \prec_0 c_\mu^\mathcal{A}$. By the above arguments, we know that this chain comprises all elements in $\mathcal{I}_0$. Moreover, due to $\mathcal{A} \models \psi_7$, this chain is the only chain satisfying the desired properties. □

Lemma 5.3.23. *For every $\ell = 1, \ldots, \kappa$ the following properties are satisfied:*

(i) *For all* $\mathsf{a}, \widetilde{\mathsf{a}} \in \mathcal{I}_\ell$ *with* $\mathcal{A}, [j \mapsto \mathsf{a}, \widetilde{j} \mapsto \widetilde{\mathsf{a}}] \models \mathrm{eq}^\ell_{j,\widetilde{j}}$ *we have* $\mathsf{a} \sim_\ell \widetilde{\mathsf{a}}$.

(ii) *All* $\mathsf{a}, \widetilde{\mathsf{a}} \in \mathcal{I}_\ell$ *with* $\mathsf{a} \sim_\ell \widetilde{\mathsf{a}}$ *satisfy* $\mathcal{A}, [j \mapsto \mathsf{a}, \widetilde{j} \mapsto \widetilde{\mathsf{a}}] \models \mathrm{eq}^\ell_{j,\widetilde{j}}$.

(iii) *For every* $f \in \mathcal{F}_{\ell-1}$ *the set* $\mathcal{S}_{\ell,f}$ *is nonempty.*

(iv) $\mathcal{I}_\ell = \bigcup_{f \in \mathcal{F}_{\ell-1}} \mathcal{S}_{\ell,f}$.

(v) *For every* $f \in \mathcal{F}_{\ell-1}$ *the set* $\mathcal{S}_{\ell,f}$ *contains exactly one element.*

(vi) *There is a unique chain* $\mathsf{a}_1 \prec_\ell \ldots \prec_\ell \mathsf{a}_p$ *comprising all elements in* $\mathcal{I}_\ell$*, and we have* $\mathcal{A} \models \mathrm{MinIdx}(\underline{\ell}, \mathsf{a}_1)$ *and* $\mathcal{A} \models \mathrm{MaxIdx}(\underline{\ell}, \mathsf{a}_p)$.

Proof. We proceed by induction on ℓ.

Base case: $\ell = 1$.

Ad (i). Due to the assumption $\mathcal{A}, [j \mapsto \mathsf{a}, \widetilde{j} \mapsto \widetilde{\mathsf{a}}] \models \mathrm{eq}^1_{j,\widetilde{j}}$, the construction of $\mathrm{eq}^1_{j,\widetilde{j}}$ entails

$$\mathcal{A} \models \bigwedge_{i=1}^{\mu} \Big(\big(J(\underline{1}, \mathsf{a}, c_i, 0) \leftrightarrow J(\underline{1}, \widetilde{\mathsf{a}}, c_i, 0) \big) \wedge \big(J(\underline{1}, \mathsf{a}, c_i, 1) \leftrightarrow J(\underline{1}, \widetilde{\mathsf{a}}, c_i, 1) \big) \Big) .$$

By Lemma 5.3.22, we know that $\mathsf{b} \in \mathcal{I}_0$ entails $\mathsf{b} = c_i^{\mathcal{A}}$ for some i. Consequently, for every $\mathsf{b} \in \mathcal{I}_0$, $\mathcal{A} \models J(\underline{1}, \mathsf{a}, \mathsf{b}, 0)$ holds if and only if $\mathcal{A} \models J(\underline{1}, \widetilde{\mathsf{a}}, \mathsf{b}, 0)$ does. This entails $\mathsf{a} \sim_\ell \widetilde{\mathsf{a}}$, because of $\mathcal{A} \models \psi_{13} \wedge \chi_2$.

Ad (ii). By definition of $\sim_1$, $\mathsf{a} \sim_1 \widetilde{\mathsf{a}}$ entails that for every $\mathsf{b} \in \mathcal{I}_0$ we have

$\mathcal{A} \models J(\underline{1}, \mathsf{a}, \mathsf{b}, 0)$ if and only if $\mathcal{A} \models J(\underline{1}, \widetilde{\mathsf{a}}, \mathsf{b}, 0)$ and
$\mathcal{A} \models J(\underline{1}, \mathsf{a}, \mathsf{b}, 1)$ if and only if $\mathcal{A} \models J(\underline{1}, \widetilde{\mathsf{a}}, \mathsf{b}, 1)$.

Moreover, Lemma 5.3.22 states that $\mathcal{I}_0 = \{c_1^{\mathcal{A}}, \ldots, c_\mu^{\mathcal{A}}\}$. Since $\mathsf{a}, \widetilde{\mathsf{a}}$ belong to $\mathcal{I}_1$, we conclude $\mathcal{A}, [j \mapsto \mathsf{a}, \widetilde{j} \mapsto \widetilde{\mathsf{a}}] \models \mathrm{eq}^1_{j,\widetilde{j}}$.

Ad (iii). $\mathcal{A} \models \psi_8$ entails $c_1^{\mathcal{A}} \prec_0 \ldots \prec_0 c_\mu^{\mathcal{A}}$. By Lemma 5.3.22, we know that this chain comprises all elements in $\mathcal{I}_0$. Moreover, due to $\mathcal{A} \models \psi_7$, this chain is the only chain satisfying this property. Hence, by Lemma 5.3.21, we can arrange all functions in $\mathcal{F}_0$ into a sequence $f_1 \sqsubset_0 \ldots \sqsubset_0 f_p$ for $p = 2^{\mu-1} + 1$. Clearly, f_1 maps every element $\mathsf{b} \in \mathcal{I}_0$ to $f_1(\mathsf{b}) = 0$, and f_p maps every element $\mathsf{b} \in \mathcal{I}_0 \setminus \{c_\mu^{\mathcal{A}}\}$ to $f_p(\mathsf{b}) = 0$ and $c_\mu^{\mathcal{A}}$ to $f_p(c_\mu^{\mathcal{A}}) = 1$. By $\mathcal{A} \models \psi_3 \wedge \psi_{10}$, we know that $\mathcal{A} \models J(\underline{1}, d_1, \mathsf{b}, 0)$ for every $\mathsf{b} \in \mathcal{I}_0$. Hence, $d_1^{\mathcal{A}} \in S_{1,f_1}$. We next show that for every k, $1 \leq k < p$, if S_{1,f_k} is nonempty, then $S_{1,f_{k+1}}$ is nonempty. Let a be an element of S_{1,f_k}. Because of $k < p$, we know that $\mathcal{A} \models J(\underline{1}, \mathsf{a}, c_\mu, 0)$. By virtue of (i) and due to $\mathcal{A} \models \psi_{16}$ we conclude that there are elements $\widetilde{\mathsf{a}}, \widetilde{\mathsf{a}}' \in \mathcal{I}_1$ such that $\mathsf{a} \sim_1 \widetilde{\mathsf{a}}$ and $\widetilde{\mathsf{a}} \prec_1 \widetilde{\mathsf{a}}'$. By Lemma 5.3.18, this results in $\widetilde{\mathsf{a}}' \in \mathcal{S}_{\ell,f_{k+1}}$.

Ad (iv). By definition of the sets $\mathcal{S}_{\ell,f}$, we have $\mathcal{S}_{1,f} \subseteq \mathcal{I}_1$. It remains to show $\mathcal{I}_1 \subseteq \bigcup_{f \in \mathcal{F}_0} \mathcal{S}_{1,f}$. As a consequence of $\mathcal{A} \models \psi_{13}$ there is a unique partial function $g_{\mathsf{a}} : \mathcal{I}_0 \to \{0, 1\}$ for every $\mathsf{a} \in \mathcal{I}_1$ such that for every $\mathsf{b} \in \mathcal{I}_0$ we have $\mathcal{A} \models J(\underline{1}, \mathsf{a}, \mathsf{b}, g_{\mathsf{a}}(\mathsf{b}))$ if and only if g_{a} is defined for b. Because of $\mathcal{A} \models \chi_2$, we know that g_{a} must be total.

For every $\mathsf{a} \in \mathcal{I}_1$ where $g_{\mathsf{a}}(c_\mu^{\mathcal{A}}) = 0$ we have $g_{\mathsf{a}} \in \mathcal{F}_0$ and thus also $\mathsf{a} \in \mathcal{S}_{1,g_{\mathsf{a}}}$.

Because of $\mathcal{A} \models \psi_5 \wedge \psi_{11} \wedge \psi_{12}$ we know that $\mathcal{A} \models \mathrm{MaxIdx}(\underline{1}, e_1)$ and that $e_1^{\mathcal{A}}$ is the only element $\mathsf{e} \in \mathcal{I}_1$ for which we have $\mathcal{A} \models J(\underline{1}, \mathsf{e}, c_\mu, 1)$. It remains to show that for every $\mathsf{b} \in \mathcal{I}_0$ with $\mathsf{b} \neq c_\mu^{\mathcal{A}}$ we have $\mathcal{A} \models J(\underline{1}, e_1, \mathsf{b}, 0)$. But this is a consequence of (iii) and the fact that the total function f_* mapping all elements b in $\mathcal{I}_0$ but $c_\mu^{\mathcal{A}}$ to $f_*(\mathsf{b}) = 0$ belongs to $\mathcal{F}_0$, and thus $\mathcal{S}_{1,f_*}$ is nonempty. In particular, $\mathcal{S}_{1,f_*} = \{e_1^{\mathcal{A}}\}$.

Consequently, $\mathcal{I}_1$ cannot contain any elements that do not lie in $\bigcup_{f \in \mathcal{F}_0} \mathcal{S}_{1,f}$.

Ad (v). Consider any set $\mathcal{S}_{1,f}$. By virtue of (iii), $\mathcal{S}_{1,f}$ contains at least one element. Suppose we are given two elements a, a' in $\mathcal{S}_{1,f}$. By virtue of Lemma 5.3.19, this means $\mathsf{a} \sim_1 \mathsf{a}'$. Because of $\mathcal{A} \models \chi_3$, there are two elements $\widetilde{\mathsf{a}}, \widetilde{\mathsf{a}}' \in \mathcal{I}_1$ for which (i) entails $\mathsf{a} \sim_1 \widetilde{\mathsf{a}}$ and $\mathsf{a}' \sim_1 \widetilde{\mathsf{a}}'$. By symmetry and transitivity of $\sim_1$, we have $\widetilde{\mathsf{a}} \sim_1 \widetilde{\mathsf{a}}'$. Hence, we observe

$$\mathcal{A} \models \forall \widetilde{i}.\, L(\underline{\ell-1}, \widetilde{i}) \rightarrow \big(J(\underline{\ell}, \widetilde{\mathsf{a}}, \widetilde{i}, 0) \leftrightarrow J(\underline{\ell}, \widetilde{\mathsf{a}}', \widetilde{i}, 0)\big)\ .$$

Consequently, $\mathcal{A} \models \chi_3$ leads to $\mathsf{a} = \mathsf{a}'$.

Ad (vi). As we have seen in the proof of Lemma 5.3.21, we can define a bijective mapping ρ that maps the functions in $\mathcal{F}_0$ to bit strings of length μ that either have a 0 as most significant bit or correspond to $10\ldots0$.

Let $p := 2^{\mu-1} + 1$. By virtue of Lemma 5.3.21, we can uniquely construct a chain

$$f_1 \sqsubset_0 f_2 \sqsubset_0 \ldots \sqsubset_0 f_p$$

comprising all functions in $\mathcal{F}_0$.

Properties (iv) and (v) together yield that $\mathcal{I}_1 = \{\mathsf{a}_1, \ldots, \mathsf{a}_p\}$ where $\mathsf{a}_k \in \mathcal{S}_{1,f_k}$ for every $k = 1, \ldots, p$. Lemma 5.3.18 says that for any $\mathsf{a}_k, \mathsf{a}_{k'}$ with $\mathsf{a}_k \prec_1 \mathsf{a}_{k'}$ we also observe $f_k \sqsubset_0 f_{k'}$. By definition of $\sqsubset_0$ and the fact that all the $f_k, f_{k'}$ are distinct, we infer that $f_k \sqsubset_0 f_{k'}$ implies $k' = k+1$ and $1 \leq k < k' \leq p$. Hence, $\mathsf{a}_k \prec_1 \mathsf{a}_{k'}$ can only hold if $k' = k+1$ and $1 \leq k < k' \leq p$.

Consider any element $\mathsf{a} \in \mathcal{I}_1$ for which $\mathcal{A} \models J(\underline{1}, \mathsf{a}, c_\mu, 0)$. By $\mathcal{A} \models \psi_{16}$, we know that there are elements $\widetilde{\mathsf{a}}, \widetilde{\mathsf{a}}' \in \mathcal{I}_1$ such that $\widetilde{\mathsf{a}} \prec_1 \widetilde{\mathsf{a}}'$ and $\mathcal{A}, [j \mapsto \mathsf{a}, \widetilde{j} \mapsto \widetilde{\mathsf{a}}] \models \mathrm{eq}^1_{j,\widetilde{j}}$. By (i), the latter translates to $\mathsf{a} \sim_1 \widetilde{\mathsf{a}}$.

Let $g, \widetilde{g}$ be functions such that $\mathsf{a} \in \mathcal{S}_{1,g}$ and $\widetilde{\mathsf{a}} \in \mathcal{S}_{1,\widetilde{g}}$. Such functions exist by virtue of (iv). However, $\mathsf{a} \sim_1 \widetilde{\mathsf{a}}$ entails $g = \widetilde{g}$, by Lemma 5.3.20. But then (v) leads to $\mathsf{a} = \widetilde{\mathsf{a}}$. Consequently, we have $\mathsf{a} \prec_1 \widetilde{\mathsf{a}}'$. By (v), this means that all but one element in $\mathcal{I}_1$ must have a $\prec_1$-successor in $\mathcal{I}_1$ and all but one elements in $\mathcal{I}_1$ are $\prec_1$-successors in $\mathcal{I}_1$. Hence, we obtain the chain $\mathsf{a}_1 \prec_1 \mathsf{a}_2 \prec_1 \ldots \prec_1 \mathsf{a}_{p-1} \prec_1 \mathsf{a}_p$ where $\mathsf{a}_1 = d_1^{\mathcal{A}}$ and $\mathsf{a}_p = e_1^{\mathcal{A}}$.

Inductive case $\ell > 1$.

Ad (i). Due to the assumption $\mathcal{A}, [j \mapsto \mathsf{a}, \widetilde{j} \mapsto \widetilde{\mathsf{a}}] \models \mathrm{eq}^\ell_{j,\widetilde{j}}$ with $\ell > 1$, the construction of $\mathrm{eq}^\ell_{j,\widetilde{j}}$ entails

$$\begin{aligned}\mathcal{A} \models\ \forall i.\, L(\underline{\ell-1}, i)\ \rightarrow\ \exists \widetilde{i}.\, L(\underline{\ell-1}, \widetilde{i}) \wedge \mathrm{eq}^{\ell-1}_{i,\widetilde{i}} \\ \wedge \Big(\big(J(\underline{\ell}, \mathsf{a}, i, 0) \leftrightarrow J(\underline{\ell}, \widetilde{\mathsf{a}}, \widetilde{i}, 0)\big) \\ \wedge \big(J(\underline{\ell}, \mathsf{a}, i, 1) \leftrightarrow J(\underline{\ell}, \widetilde{\mathsf{a}}, \widetilde{i}, 1)\big)\Big)\ .\end{aligned}$$

By inductive application of (i), $\mathcal{A}, [i \mapsto \mathsf{b}, \widetilde{i} \mapsto \widetilde{\mathsf{b}}] \models \mathrm{eq}^{\ell-1}_{i,\widetilde{i}}$ entails $\mathsf{b} \sim_{\ell-1} \widetilde{\mathsf{b}}$. Inductive application of (iv) implies that $\mathsf{b} \in \mathcal{S}_{\ell-1,f}$ for some $f \in \mathcal{F}_{\ell-1}$. By Lemma 5.3.20 together with $\mathsf{b} \sim_{\ell-1} \widetilde{\mathsf{b}}$, we conclude $\widetilde{\mathsf{b}} \in \mathcal{S}_{\ell-1,f}$. Now, inductive application of (v) leads to $\mathsf{b} = \widetilde{\mathsf{b}}$. This means we in fact have

$$\begin{aligned}\mathcal{A} \models\ \forall i.\, L(\underline{\ell-1}, i) \rightarrow \Big(\big(J(\underline{\ell}, \mathsf{a}, i, 0) \leftrightarrow J(\underline{\ell}, \widetilde{\mathsf{a}}, i, 0)\big) \\ \wedge \big(J(\underline{\ell}, \mathsf{a}, i, 1) \leftrightarrow J(\underline{\ell}, \widetilde{\mathsf{a}}, i, 1)\big)\Big)\ .\end{aligned}$$

In other words, $\mathsf{a} \sim_\ell \widetilde{\mathsf{a}}$.

Ad (ii). By definition of $\sim_\ell$, $\mathsf{a} \sim_\ell \widetilde{\mathsf{a}}$ entails that for every $\mathsf{b} \in \mathcal{I}_{\ell-1}$ we have

$\mathcal{A} \models J(\underline{1}, \mathsf{a}, \mathsf{b}, 0)$ if and only if $\mathcal{A} \models J(\underline{1}, \widetilde{\mathsf{a}}, \mathsf{b}, 0)$ and

$\mathcal{A} \models J(\underline{1}, \mathsf{a}, \mathsf{b}, 1)$ if and only if $\mathcal{A} \models J(\underline{1}, \widetilde{\mathsf{a}}, \mathsf{b}, 1)$.

By (i) and the fact that $\sim_{\ell-1}$ is an equivalence relation and thus $\mathsf{b} \sim_{\ell-1} \mathsf{b}$, we conclude $\mathcal{A}, [i{\mapsto}\mathsf{b}, \widetilde{i}{\mapsto}\mathsf{b}] \models \mathrm{eq}^{\ell-1}_{i,\widetilde{i}}$ for every $\mathsf{b} \in \mathcal{I}_{\ell-1}$. Consequently, $\mathcal{A}, [j{\mapsto}\mathsf{a}, \widetilde{j}{\mapsto}\widetilde{\mathsf{a}}] \models \mathrm{eq}^{\ell}_{j,\widetilde{j}}$.

Ad (iii). By inductive application of (vi), we know that there is a unique chain $\mathsf{b}_1 \prec_{\ell-1} \ldots \prec_{\ell-1} \mathsf{b}_p$ comprising all elements in $\mathcal{I}_{\ell-1}$. Moreover, we observe $\mathcal{A} \models \mathrm{MinIdx}(\underline{\ell-1}, \mathsf{b}_1)$ and $\mathcal{A} \models \mathrm{MaxIdx}(\underline{\ell-1}, \mathsf{b}_p)$. Hence, by Lemma 5.3.21, we can arrange all mappings in $\mathcal{F}_{\ell-1}$ into a sequence $f_1 \sqsubset_{\ell-1} \ldots \sqsubset_{\ell-1} f_{p'}$ where $p' = 2^{p-1} + 1$.

Clearly, f_1 maps every element $\mathsf{b} \in \mathcal{I}_{\ell-1}$ to $f_1(\mathsf{b}) = 0$, and $f_{p'}$ maps every element b_k with $k < p$ to $f_{p'}(\mathsf{b}_k) = 0$ and b_p to $f_{p'}(\mathsf{b}_p) = 1$. By $\mathcal{A} \models \psi_3 \wedge \psi_{10}$, we know that $\mathcal{A} \models J(\underline{\ell}, d_\ell, \mathsf{b}, 0)$ for every $\mathsf{b} \in \mathcal{I}_{\ell-1}$. Hence, $d^{\mathcal{A}}_\ell \in S_{\ell,f_1}$.

We next show that for every k, $1 \leq k < p'$, if S_{ℓ,f_k} is nonempty, then $S_{\ell,f_{k+1}}$ is nonempty. Let a be an element of S_{ℓ,f_k}. Because of $k < p'$, we know that $\mathcal{A} \models J(\underline{\ell}, \mathsf{a}, \mathsf{b}_p, 0)$. By virtue of (i) and due to $\mathcal{A} \models \psi_{16}$ we conclude that there are elements $\widetilde{\mathsf{a}}, \widetilde{\mathsf{a}}' \in \mathcal{I}_\ell$ such that $\mathsf{a} \sim_\ell \widetilde{\mathsf{a}}$ and $\widetilde{\mathsf{a}} \prec_\ell \widetilde{\mathsf{a}}'$. Moreover, Lemma 5.3.20 leads to $\widetilde{\mathsf{a}} \in S_{\ell,f_k}$. By Lemma 5.3.18, this results in $\widetilde{\mathsf{a}}' \in \mathcal{S}_{\ell,f_{k+1}}$.

Ad (iv). We have $\bigcup_{f \in \mathcal{F}_{\ell-1}} \mathcal{S}_{\ell,f} \subseteq \mathcal{I}_\ell$ by definition of the sets $\mathcal{S}_{\ell,f}$. It thus remains to show $\mathcal{I}_\ell \subseteq \bigcup_{f \in \mathcal{F}_{\ell-1}} \mathcal{S}_{\ell,f}$. As a consequence of $\mathcal{A} \models \psi_{13}$ there is a unique partial mapping $g_{\mathsf{a}} : \mathcal{I}_{\ell-1} \to \{0,1\}$ for every $\mathsf{a} \in \mathcal{I}_\ell$ such that for every $\mathsf{b} \in \mathcal{I}_{\ell-1}$ we have $\mathcal{A} \models J(\underline{\ell}, \mathsf{a}, \mathsf{b}, g_{\mathsf{a}}(\mathsf{b}))$ if and only if g_{a} is defined for b. Because of $\mathcal{A} \models \chi_2$, we know that g_{a} must be total.

For every $\mathsf{a} \in \mathcal{I}_\ell$ where $g_{\mathsf{a}}(e^{\mathcal{A}}_{\ell-1}) = 0$ we have $g_{\mathsf{a}} \in \mathcal{F}_{\ell-1}$ and thus also $\mathsf{a} \in \mathcal{S}_{\ell,g_{\mathsf{a}}}$.

Because of $\mathcal{A} \models \psi_5 \wedge \psi_{11} \wedge \psi_{12}$ we know that $\mathcal{A} \models \mathrm{MaxIdx}(\underline{\ell}, e_\ell)$ and that $e^{\mathcal{A}}_\ell$ is the only element $\mathsf{e} \in \mathcal{I}_\ell$ for which we have $\mathcal{A} \models J(\underline{\ell}, \mathsf{e}, e_{\ell-1}, 1)$. It remains to show that for every $\mathsf{b} \in \mathcal{I}_{\ell-1}$ with $\mathsf{b} \neq e^{\mathcal{A}}_{\ell-1}$ we have $\mathcal{A} \models J(\underline{\ell}, e_\ell, \mathsf{b}, 0)$. But this is a consequence of (iii) and the fact that the total function f_* mapping all elements b in $\mathcal{I}_{\ell-1}$ but $e^{\mathcal{A}}_{\ell-1}$ to $f_*(\mathsf{b}) = 0$ belongs to $\mathcal{F}_{\ell-1}$, and thus $\mathcal{S}_{\ell,f_*}$ is nonempty. In particular, $\mathcal{S}_{\ell,f_*} = \{e^{\mathcal{A}}_\ell\}$.

Consequently, $\mathcal{I}_\ell$ cannot contain any elements that do not lie in $\bigcup_{f \in \mathcal{F}_{\ell-1}} \mathcal{S}_{\ell,f}$.

Ad (v). Consider any set $\mathcal{S}_{\ell,f}$. By virtue of (iii), $\mathcal{S}_{\ell,f}$ contains at least one element. Suppose we are given two elements a, a' in $\mathcal{S}_{\ell,f}$. By virtue of Lemma 5.3.19, this means $\mathsf{a} \sim_\ell \mathsf{a}'$. Because of $\mathcal{A} \models \chi_3$, there are two elements $\widetilde{\mathsf{a}}, \widetilde{\mathsf{a}}' \in \mathcal{I}_\ell$ for which (i) entails $\mathsf{a} \sim_\ell \widetilde{\mathsf{a}}$ and $\mathsf{a}' \sim_\ell \widetilde{\mathsf{a}}'$. By symmetry and transitivity of $\sim_\ell$, we have $\widetilde{\mathsf{a}} \sim_\ell \widetilde{\mathsf{a}}'$. Hence, we observe

$$\mathcal{A} \models \forall \widetilde{i}.\, L(\underline{\ell-1}, \widetilde{i}) \rightarrow \big(J(\underline{\ell}, \widetilde{\mathsf{a}}, \widetilde{i}, 0) \leftrightarrow J(\underline{\ell}, \widetilde{\mathsf{a}}', \widetilde{i}, 0)\big)\ .$$

Consequently, $\mathcal{A} \models \chi_3$ leads to $\mathsf{a} = \mathsf{a}'$.

Ad (vi). As we have seen in the proof of Lemma 5.3.21, we can define a bijective mapping ρ that maps the functions in $\mathcal{F}_{\ell-1}$ to bit strings of length μ that either have a 0 as most significant bit or correspond to $10\ldots0$.

Let $p' := 2^{|\mathcal{I}_{\ell-1}|-1} + 1$. By virtue of Lemma 5.3.21, we can uniquely construct a chain

$$f_1 \sqsubset_{\ell-1} f_2 \sqsubset_{\ell-1} \ldots \sqsubset_{\ell-1} f_{p'}$$

comprising all functions in $\mathcal{F}_{\ell-1}$.

Properties (iv) and (v) together yield that $\mathcal{I}_\ell = \{\mathsf{a}_1, \ldots, \mathsf{a}_{p'}\}$ where $\mathsf{a}_k \in \mathcal{S}_{\ell,f_k}$ for every $k = 1, \ldots, p'$. Lemma 5.3.18 says that for any $\mathsf{a}_k, \mathsf{a}_{k'}$ with $\mathsf{a}_k \prec_\ell \mathsf{a}_{k'}$ we observe $f_k \sqsubset_{\ell-1} f_{k'}$. By definition of $\sqsubset_{\ell-1}$ and the fact that all the $f_k, f_{k'}$ are distinct, $f_k \sqsubset_{\ell-1} f_{k'}$ implies $k' = k+1$ and $1 \leq k < k' \leq p'$. Hence, $\mathsf{a}_k \prec_\ell \mathsf{a}_{k'}$ can only hold if $k' = k+1$ and $1 \leq k < k' \leq p'$.

Consider any element $\mathsf{a} \in \mathcal{I}_\ell$ for which $\mathcal{A} \models J(\underline{\ell}, \mathsf{a}, e_{\ell-1}, 0)$. By $\mathcal{A} \models \psi_{16}$, we know that there are elements $\widetilde{\mathsf{a}}, \widetilde{\mathsf{a}}' \in \mathcal{I}_\ell$ such that $\widetilde{\mathsf{a}} \prec_\ell \widetilde{\mathsf{a}}'$ and $\mathcal{A}, [j{\mapsto}\mathsf{a}, \widetilde{j}{\mapsto}\widetilde{\mathsf{a}}] \models \mathrm{eq}^{\ell}_{j,\widetilde{j}}$. By (i), the latter translates to $\mathsf{a} \sim_\ell \widetilde{\mathsf{a}}$.

Let $g, \widetilde{g}$ be functions such that $\mathsf{a} \in \mathcal{S}_{\ell,g}$ and $\widetilde{\mathsf{a}} \in \mathcal{S}_{\ell,\widetilde{g}}$. Such functions exist by virtue of (iv). However, $\mathsf{a} \sim_\ell \widetilde{\mathsf{a}}$ entails $g = \widetilde{g}$, by Lemma 5.3.20. Moreover, (v) leads to $\mathsf{a} = \widetilde{\mathsf{a}}$. Consequently, we have $\mathsf{a} \prec_\ell \widetilde{\mathsf{a}}'$. By (v), this means that all but one element in $\mathcal{I}_\ell$ must have a successor in $\mathcal{I}_\ell$ and all but one element in $\mathcal{I}_\ell$ are successor in $\mathcal{I}_\ell$. Hence, we obtain the chain $\mathsf{a}_1 \prec_\ell \mathsf{a}_2 \prec_\ell \ldots \prec_\ell \mathsf{a}_{p'-1} \prec_\ell \mathsf{a}_{p'}$ where $\mathsf{a}_1 = d_\ell^{\mathcal{A}}$ and $\mathsf{a}_{p'} = e_\ell^{\mathcal{A}}$. □

5.3.2 Formalizing a Tiling of a Torus

In order to formalize a given domino system $\mathfrak{D} = \langle \mathcal{D}, \mathcal{H}, \mathcal{V} \rangle$ and an initial condition $\overline{D}$, we introduce the following constant symbols and predicate symbols:

$H(x,y,x',y')$	$\langle x', y' \rangle$ is the horizontal neighbor of $\langle x, y \rangle$, i.e. $x' = x + 1 \pmod{2^{\uparrow\kappa}(\mu-1)+1}$ and $y' = y$,
$V(x,y,x',y')$	$\langle x', y' \rangle$ is the vertical neighbor of $\langle x, y \rangle$,
$\underline{D}(x,y)$	$\langle x, y \rangle$ is tiled with $D \in \mathcal{D}$,
$f_1, \ldots, f_{\lvert\overline{D}\rvert}$	constant symbols addressing the points $\langle 0,0\rangle, \ldots, \langle \lvert\overline{D}\rvert - 1, 0\rangle$.

With the ideas we have used to formalize the index hierarchy in the previous section, it is now fairly simple to formalize the torus. The following sentences encode a given domino system $\mathfrak{D} := \langle \mathcal{D}, \mathcal{H}, \mathcal{V} \rangle$ plus an initial condition $\overline{D}$, which is a finite word over $\mathcal{D}$. We try to make as many sentences as possible equivalent to Horn sentences. Brief descriptions of the intended meaning are added for selected parts.

$$\eta_1 := \forall xyx'y'.\ H(x,y,x',y') \ \rightarrow\ L(\underline{\kappa},x) \wedge L(\underline{\kappa},y) \wedge L(\underline{\kappa},x') \wedge L(\underline{\kappa},y') \wedge y \approx y'$$

$$\eta_2 := \forall xyx'y'i.\ H(x,y,x',y') \wedge \mathrm{MaxIdx}(\underline{\kappa-1},i) \wedge J(\underline{\kappa},x,i,0) \ \rightarrow\ \mathrm{Succ}(\underline{\kappa},x,x')$$

$$\eta_3 := \forall xyi.\ L(\underline{\kappa},x) \wedge L(\underline{\kappa},y) \wedge \mathrm{MaxIdx}(\underline{\kappa-1},i) \wedge J(\underline{\kappa},x,i,0) \rightarrow\ \exists\widetilde{x}.\, \mathrm{eq}^{\kappa}_{x,\widetilde{x}} \wedge \exists\widetilde{y}.\, \mathrm{eq}^{\kappa}_{y,\widetilde{y}} \wedge \Big(\bigwedge_{D\in\mathcal{D}} \underline{D}(x,y) \leftrightarrow \underline{D}(\widetilde{x},\widetilde{y}) \Big) \wedge \exists\widetilde{x}'.\, H(\widetilde{x},\widetilde{y},\widetilde{x}',\widetilde{y})$$

η_3 ensures that every point not on the horizontal "edge" of the torus has a horizontal neighbor.

$$\eta_4 := \forall xyx'.\ \mathrm{MaxIdx}(\underline{\kappa},x) \wedge \mathrm{MinIdx}(\underline{\kappa},x') \ \rightarrow\ H(x,y,x',y)$$

$$\eta_5 := \forall xyx'y'.\ H(x,y,x',y') \wedge \mathrm{MaxIdx}(\underline{\kappa},x) \ \rightarrow\ \mathrm{MinIdx}(\underline{\kappa},x')$$

$$\eta_6 := \forall xyx'y'.\ H(x,y,x',y') \wedge \mathrm{MinIdx}(\underline{\kappa},x') \ \rightarrow\ \mathrm{MaxIdx}(\underline{\kappa},x)$$

η_4 to η_6 establish the points $\langle 2^{\uparrow\kappa}(\mu-1)+1, y\rangle$ and $\langle 0, y\rangle$ as horizontal neighbors for every y.

$$\eta_7 := \forall xyx'y'.\ V(x,y,x',y') \ \rightarrow\ L(\underline{\kappa},x) \wedge L(\underline{\kappa},y) \wedge L(\underline{\kappa},x') \wedge L(\underline{\kappa},y') \wedge x \approx x'$$

$$\eta_8 := \forall xyx'y'.\ V(x,y,x',y') \wedge \mathrm{MaxIdx}(\underline{\kappa-1},i) \wedge J(\underline{\kappa},y,i,0) \ \rightarrow\ \mathrm{Succ}(\underline{\kappa},y,y')$$

$$\eta_9 := \forall xyi.\ L(\underline{\kappa},x) \wedge L(\underline{\kappa},y) \wedge \mathrm{MaxIdx}(\underline{\kappa-1},i) \wedge J(\underline{\kappa},y,i,0) \rightarrow\ \exists\widetilde{x}.\, \mathrm{eq}^{\kappa}_{x,\widetilde{x}} \wedge \exists\widetilde{y}.\, \mathrm{eq}^{\kappa}_{y,\widetilde{y}} \wedge \Big(\bigwedge_{D\in\mathcal{D}} \big(\underline{D}(x,y) \leftrightarrow \underline{D}(\widetilde{x},\widetilde{y})\big) \Big) \wedge \exists\widetilde{y}'.\, V(\widetilde{x},\widetilde{y},\widetilde{x},\widetilde{y}')$$

η_9 ensures that every point not on the vertical "edge" of the torus has a vertical neighbor.

$$\eta_{10} := \forall xyy'.\ \mathrm{MaxIdx}(\underline{\kappa},y) \wedge \mathrm{MinIdx}(\underline{\kappa},y') \ \rightarrow\ V(x,y,x,y')$$

$$\eta_{11} := \forall xx'yy'.\ V(x,y,x',y') \wedge \mathrm{MaxIdx}(\underline{\kappa},y) \ \rightarrow\ \mathrm{MinIdx}(\underline{\kappa},y')$$

$$\eta_{12} := \forall xx'yy'.\ V(x,y,x',y') \wedge \mathrm{MinIdx}(\underline{\kappa},y') \ \rightarrow\ \mathrm{MaxIdx}(\underline{\kappa},y)$$

η_{10} to η_{12} establish the points $\langle x, 2^{\uparrow\kappa}(\mu-1)+1\rangle$ and $\langle x, 0\rangle$ as vertical neighbors for every x.

$$\eta_{13} := \bigwedge_{D\in\mathcal{D}} \forall xy.\ \underline{D}(x,y) \ \rightarrow\ L(\underline{\kappa},x) \wedge L(\underline{\kappa},y)$$

$$\eta_{14} := \bigwedge_{D\in\mathcal{D}} \bigwedge_{D'\in\mathcal{D}\setminus\{D\}} \forall xy.\ \underline{D}(x,y) \ \rightarrow\ \neg\underline{D}'(x,y)$$

$$\eta_{15} := \forall xx'y.\ H(x,y,x',y) \ \rightarrow \bigvee_{\langle D,D'\rangle\in\mathcal{H}} \underline{D}(x,y) \wedge \underline{D}'(x',y)$$

$$\eta_{16} := \forall xyy'.\, V(x,y,x,y') \;\rightarrow \bigvee_{\langle D,D'\rangle \in \mathcal{V}} \underline{D}(x,y) \wedge \underline{D}'(x,y')$$

η_{15} and η_{16} ensure that the rules of the domino system $\mathfrak{D}$ are obeyed. These are the only essentially non-Horn sentences among η_1 to η_{28}.

$$\eta_{17} := \forall z.\, \mathrm{MinIdx}(\underline{\kappa}, z) \;\rightarrow\; f_1 \approx z \wedge \bigwedge_{i=1}^{n-1} H(f_i, z, f_{i+1}, z)$$

$$\eta_{18} := \forall z.\, \mathrm{MinIdx}(\underline{\kappa}, z) \;\rightarrow\; \bigwedge_{i=1}^{n} \underline{D}_i(f_i, z)$$

η_{17} and η_{18} express the initial condition $\overline{D}$, i.e. the lower left domino tiles are predefined to be the sequence $\overline{D} = D_1 \dots D_n$.

Regarding the length of the sentences $\eta_1, \dots, \eta_{18}$, we observe the following:

$$\mathrm{len}(\eta_1), \mathrm{len}(\eta_2), \mathrm{len}(\eta_4), \dots, \mathrm{len}(\eta_8), \mathrm{len}(\eta_{10}), \dots, \mathrm{len}(\eta_{12}) \in \mathcal{O}(\log \kappa),$$

$$\mathrm{len}(\eta_3), \mathrm{len}(\eta_9) \in \mathcal{O}\big(\kappa \log \kappa + \mu(\log \kappa + \log \mu) + |\mathcal{D}| \log |\mathcal{D}|\big),$$

$$\mathrm{len}(\eta_{13}) \in \mathcal{O}\big(|\mathcal{D}|(\log |\mathcal{D}| + \log \kappa)\big),$$

$$\mathrm{len}(\eta_{14}) \in \mathcal{O}\big(|\mathcal{D}|^2 \log |\mathcal{D}|\big),$$

$$\mathrm{len}(\eta_{15}) \in \mathcal{O}\big(|\mathcal{H}| \log |\mathcal{D}|\big) = \mathcal{O}\big(|\mathcal{D}|^2 \log |\mathcal{D}|\big),$$

$$\mathrm{len}(\eta_{16}) \in \mathcal{O}\big(|\mathcal{V}| \log |\mathcal{D}|\big) = \mathcal{O}\big(|\mathcal{D}|^2 \log |\mathcal{D}|\big),$$

$$\mathrm{len}(\eta_{17}) \in \mathcal{O}\big(\log \kappa + n \log n\big),$$

$$\mathrm{len}(\eta_{18}) \in \mathcal{O}\big(\log \kappa + n(\log n + \log |\mathcal{D}|)\big).$$

In total, the length of $\eta_1 \wedge \ldots \wedge \eta_{18}$ lies in $\mathcal{O}\big(\widehat{n} \log \widehat{n}\big)$, where $\widehat{n} := \max\{\kappa, \mu, n, |\mathcal{D}|^2\}$.

Next, we show correctness of the formalization. More concretely, we aim to prove that any model of $\psi_1 \wedge \ldots \wedge \psi_{16} \wedge \chi_1 \wedge \chi_2 \wedge \chi_3 \wedge \eta_1 \wedge \ldots \wedge \eta_{18}$ induces a tiling τ of $\mathbb{Z}_r^2$ for $r := 2^{\uparrow\kappa}(\mu - 1) + 1$ and with initial condition $\overline{D} := D_1, \dots, D_n$. This will be the statement of Lemma 5.3.27.

$\mathcal{A}$ For the remainder of this subsection we assume that $\mathcal{A}$ indeed satisfies the sentence $\psi_1 \wedge \ldots \wedge \psi_{16} \wedge \chi_1 \wedge \chi_2 \wedge \chi_3 \wedge \eta_1 \wedge \ldots \wedge \eta_{18}$. Moreover, we take over the notation from Lemma 5.3.14 and $\mathcal{I}_\ell$, $\prec_\ell$, $\sim_\ell$ Definition 5.3.15, e.g. $\mathcal{I}_\ell$, $\prec_\ell$, and $\sim_\ell$. In addition, we define the following relations:

$\prec^H$ $\prec^H \subseteq \mathcal{I}_\kappa^2 \times \mathcal{I}_\kappa^2$ such that $\langle \mathsf{a}, \mathsf{b}\rangle \prec^H \langle \mathsf{a}', \mathsf{b}'\rangle$ if and only if $\mathcal{A} \models H(\mathsf{a}, \mathsf{b}, \mathsf{a}', \mathsf{b}')$;

$\prec^V$ $\prec^V \subseteq \mathcal{I}_\kappa^2 \times \mathcal{I}_\kappa^2$ such that $\langle \mathsf{a}, \mathsf{b}\rangle \prec^V \langle \mathsf{a}', \mathsf{b}'\rangle$ if and only if $\mathcal{A} \models V(\mathsf{a}, \mathsf{b}, \mathsf{a}', \mathsf{b}')$.

Roughly speaking, we will show in Lemmas 5.3.24 and 5.3.25 that $\mathcal{I}_\kappa \times \mathcal{I}_\kappa$ together with the horizontal and vertical neighborhood relations is isomorphic to the torus $\mathbb{Z}_r^2$ for $r = 2^{\uparrow\kappa}(\mu - 1) + 1$. Lemma 5.3.26 states that under $\mathcal{A}$ for each pair in $\langle \mathsf{a}, \mathsf{b}\rangle \in \mathcal{I}_\kappa \times \mathcal{I}_\kappa$ there is exactly one tile $D \in \mathcal{D}$ assigned to $\langle \mathsf{a}, \mathsf{b}\rangle$. Finally, Lemma 5.3.27 states that $\mathcal{A}$ induces a tiling which starts from the initial condition $\overline{D}$.

Lemma 5.3.24. *For all pairs $\langle \mathsf{a}, \mathsf{b}\rangle, \langle \mathsf{a}', \mathsf{b}'\rangle \in \mathcal{I}_\kappa^2$ we observe the following properties.*

(i) $\langle \mathsf{a}, \mathsf{b}\rangle \prec^H \langle \mathsf{a}', \mathsf{b}'\rangle$ *entails that* $\mathsf{b} = \mathsf{b}'$ *and that either* $\mathsf{a} \prec_\kappa \mathsf{a}'$, *or* $\mathsf{a} = e_\kappa^{\mathcal{A}}$ *and* $\mathsf{a}' = d_\kappa^{\mathcal{A}}$.

(ii) $\mathsf{a} = e_\kappa^{\mathcal{A}}$ *and* $\mathsf{a}' = d_\kappa^{\mathcal{A}}$ *entails* $\langle \mathsf{a}, \mathsf{b}\rangle \prec^H \langle \mathsf{a}', \mathsf{b}\rangle$ *for every* $\mathsf{b} \in \mathcal{I}_\kappa$.

(iii) $\mathsf{a} \prec_\kappa \mathsf{a}'$ *implies* $\langle \mathsf{a}, \mathsf{b}\rangle \prec^H \langle \mathsf{a}', \mathsf{b}\rangle$ *for every* $\mathsf{b} \in \mathcal{I}_\kappa$.

(iv) $\langle \mathsf{a}, \mathsf{b}\rangle \prec^V \langle \mathsf{a}', \mathsf{b}'\rangle$ *entails that* $\mathsf{a} = \mathsf{a}'$ *and that either* $\mathsf{b} \prec_\kappa \mathsf{b}'$ *or* $\mathsf{b} = e_\kappa^{\mathcal{A}}$ *and* $\mathsf{b}' = d_\kappa^{\mathcal{A}}$.

(v) $\mathsf{b} = e_\kappa^{\mathcal{A}}$ *and* $\mathsf{b}' = d_\kappa^{\mathcal{A}}$ *entails* $\langle \mathsf{a}, \mathsf{b}\rangle \prec^V \langle \mathsf{a}, \mathsf{b}'\rangle$ *for every* $\mathsf{a} \in \mathcal{I}_\kappa$.

(vi) $\mathsf{b} \prec_\kappa \mathsf{b}'$ *implies* $\langle \mathsf{a}, \mathsf{b}\rangle \prec^V \langle \mathsf{a}, \mathsf{b}'\rangle$ *for every* $\mathsf{a} \in \mathcal{I}_\kappa$.

Proof. Property (i) follows by $\mathcal{A} \models \eta_1 \wedge \eta_2 \wedge \eta_5 \wedge \eta_6$. Property (ii) follows by $\mathcal{A} \models \eta_4$.

In order to show Property (iii), we have to argue a bit more. First of all, we conclude $\mathsf{a} \neq e_\kappa^{\mathcal{A}}$, by $\mathcal{A} \models \psi_2 \wedge \psi_4$. Because of $\mathcal{A} \models \eta_3$ and Lemma 5.3.23(i), for all a, b with $\mathsf{a} \neq e_\kappa^{\mathcal{A}}$ there must exist $\widetilde{\mathsf{a}}, \widetilde{\mathsf{b}}, \widetilde{\mathsf{a}}' \in \mathcal{I}_\kappa$ such that $\mathsf{a} \sim_\kappa \widetilde{\mathsf{a}}$, $\mathsf{b} \sim_\kappa \widetilde{\mathsf{b}}$, and $\langle \widetilde{\mathsf{a}}, \widetilde{\mathsf{b}}\rangle \prec^H \langle \widetilde{\mathsf{a}}', \widetilde{\mathsf{b}}\rangle$. Due to Lemma 5.3.20 in combination with Lemma 5.3.23(v), we get $\mathsf{a} = \widetilde{\mathsf{a}}$ and $\mathsf{b} = \widetilde{\mathsf{b}}$. Thus, we have $\langle \mathsf{a}, \mathsf{b}\rangle \prec^H \langle \widetilde{\mathsf{a}}', \mathsf{b}\rangle$. Moreover, $\mathsf{a} \neq e_\kappa^{\mathcal{A}}$ together with (i) leads to $\mathsf{a} \prec_\kappa \widetilde{\mathsf{a}}'$. Since we assumed $\mathsf{a} \prec_\kappa \mathsf{a}'$, Lemma 5.3.23(vi) says that a' is the only element satisfying $\mathsf{a} \prec_\kappa \mathsf{a}'$, i.e. $\widetilde{\mathsf{a}}' = \mathsf{a}'$. Consequently, we have $\langle \mathsf{a}, \mathsf{b}\rangle \prec^H \langle \mathsf{a}', \mathsf{b}\rangle$.

Properties (iv) to (vi) can be proved analogously to the first three properties using $\mathcal{A} \models \eta_7 \wedge \ldots \wedge \eta_{12}$. □

Lemma 5.3.25. *Let* $r := 2^{\uparrow\kappa}(\mu - 1) + 1$. *There is a bijective mapping* $\rho : \mathbb{Z}_r^2 \to \mathcal{I}_\kappa^2$ *such that* $\rho(0,0) = \langle d_\kappa^{\mathcal{A}}, d_\kappa^{\mathcal{A}}\rangle$ *and for every pair* $\langle s, t\rangle \in \mathbb{Z}_r^2$ *we have* $\rho(s,t) \prec^H \rho(s+1,t)$ *and* $\rho(s,t) \prec^V \rho(s,t+1)$ *where* $+$ *stands for addition modulo* r.

Proof. By Lemma 5.3.23 we know that there is a unique chain $\mathsf{a}_1 \prec_\kappa \ldots \prec_\kappa \mathsf{a}_r$ comprising all elements in $\mathcal{I}_\kappa$. Notice that $\mathsf{a}_k, \mathsf{a}_{k'}$ with $k \neq k'$ are distinct. We define ρ so that $\rho(s,t) := \langle \mathsf{a}_{s+1}, \mathsf{a}_{t+1}\rangle$ for all $s, t \in \mathbb{Z}_r = \{0, \ldots r-1\}$.

Obviously, ρ is bijective. Since a_1 is the only element in $\mathcal{I}_\kappa$ for which there is no b in the above chain with $\mathsf{b} \prec_\kappa \mathsf{a}_1$, $\mathcal{A} \models \psi_2 \wedge \psi_3$ enforces $\mathsf{a}_1 = d_\kappa^{\mathcal{A}}$. Hence, $\rho(0,0) = \langle d_\kappa^{\mathcal{A}}, d_\kappa^{\mathcal{A}}\rangle$.

Since a_r is the only element in $\mathcal{I}_\kappa$ for which there is no b' in the above chain with $\mathsf{a}_r \prec_\kappa \mathsf{b}'$, $\mathcal{A} \models \psi_4 \wedge \psi_5$ enforces $\mathsf{a}_r = e_\kappa^{\mathcal{A}}$. Hence, Lemma 5.3.24(ii) entails $\rho(r-1,t) \prec^H \rho(0,t)$ for every $t \in \mathbb{Z}_r$. Moreover, the existence of the above chain together with Lemma 5.3.24(iii) leads to $\rho(s,t) \prec^H \rho(s+1,t)$ for every $s \in \mathbb{Z}_r \setminus \{r-1\}$ and every $t \in \mathbb{Z}_r$. Consequently, we observe $\rho(s,t) \prec^H \rho(s+1,t)$ — modulo r — for every pair $\langle s, t\rangle \in \mathbb{Z}_r^2$.

By similar arguments, we infer $\rho(s,t) \prec^V \rho(s,t+1)$ for every pair $\langle s, t\rangle \in \mathbb{Z}_r^2$. □

Lemma 5.3.26. *Suppose that* $\mathcal{D}$, $\mathcal{H}$, *and* $\mathcal{V}$ *are nonempty. For all pairs* $\langle \mathsf{a}, \mathsf{b}\rangle \in \mathcal{I}_\kappa^2$ *we have* $\mathcal{A} \models \underline{D}(\mathsf{a}, \mathsf{b})$ *for exactly one* $D \in \mathcal{D}$.

Proof. Due to $\mathcal{A} \models \eta_{15} \wedge \eta_{16}$, we observe the following properties for all pairs $\langle \mathsf{a}, \mathsf{b}\rangle, \langle \mathsf{a}', \mathsf{b}'\rangle \in \mathcal{I}_\kappa^2$:

$\langle \mathsf{a}, \mathsf{b}\rangle \prec^H \langle \mathsf{a}', \mathsf{b}'\rangle$ implies that there are $D, D' \in \mathcal{D}$ such that $\langle D, D'\rangle \in \mathcal{H}$ and $\mathcal{A} \models \underline{D}(\mathsf{a}, \mathsf{b})$ and $\mathcal{A} \models \underline{D}'(\mathsf{a}', \mathsf{b}')$;

$\langle \mathsf{a}, \mathsf{b}\rangle \prec^V \langle \mathsf{a}', \mathsf{b}'\rangle$ implies that there are $D, D' \in \mathcal{D}$ such that $\langle D, D'\rangle \in \mathcal{V}$ and $\mathcal{A} \models \underline{D}(\mathsf{a}, \mathsf{b})$ and $\mathcal{A} \models \underline{D}'(\mathsf{a}', \mathsf{b}')$.

By virtue of Lemma 5.3.25, we know that there is a bijection ρ such that for every pair $\langle \mathsf{a}, \mathsf{b}\rangle$ in the image of ρ there is another pair $\langle \mathsf{a}', \mathsf{b}'\rangle$ such that $\langle \mathsf{a}, \mathsf{b}\rangle \prec^H \langle \mathsf{a}', \mathsf{b}'\rangle$ or $\langle \mathsf{a}, \mathsf{b}\rangle \prec^V \langle \mathsf{a}', \mathsf{b}'\rangle$. Since the image of ρ covers the entire set $\mathcal{I}_\kappa^2$, this means that there is at least one $D \in \mathcal{D}$ for every pair $\langle \mathsf{a}, \mathsf{b}\rangle \in \mathcal{I}_\kappa^2$ such that $\mathcal{A} \models \underline{D}(\mathsf{a}, \mathsf{b})$. Finally, because of $\mathcal{A} \models \eta_{14}$ we know that there is at most one $D \in \mathcal{D}$ for every pair $\langle \mathsf{a}, \mathsf{b}\rangle \in \mathcal{I}_\kappa^2$ such that $\mathcal{A} \models \underline{D}(\mathsf{a}, \mathsf{b})$. □

Lemma 5.3.27. *Let* $r := 2^{\uparrow\kappa}(\mu - 1) + 1$ *and assume that* $\mathcal{D}$, $\mathcal{H}$, *and* $\mathcal{V}$ *are nonempty.* $\mathcal{A}$ *induces a tiling* τ *of* $\mathbb{Z}_r^2$ *with initial condition* $\overline{D} := D_1, \ldots, D_n$.

Proof. Let ρ be a bijection according to Lemma 5.3.25. We define the mapping τ such that $\tau(s,t) := D$ if and only if $\rho(s,t) = \langle \mathsf{a}, \mathsf{b}\rangle$ and $\mathcal{A} \models \underline{D}(\mathsf{a}, \mathsf{b})$. By Lemma 5.3.26, we know that τ is well defined. By $\mathcal{A} \models \eta_{17} \wedge \eta_{18}$, we know that $\mathcal{A} \models \underline{D}_i(\mathsf{a}, \mathsf{b})$ for $\langle \mathsf{a}, \mathsf{b}\rangle = \rho(i, 0)$ and $i = 0, \ldots, n-1$. Hence, τ satisfies the initial condition. By definition of ρ and because of $\mathcal{A} \models \eta_{15} \wedge \eta_{16}$, we observe the following:

(a) For every pair $\langle s, t\rangle \in \mathbb{Z}_r^2$ there are pairs $\langle D, D'\rangle \in \mathcal{H}$, $\langle \mathsf{a}, \mathsf{b}\rangle = \rho(s,t)$, and $\langle \mathsf{a}', \mathsf{b}'\rangle = \rho(s+1,t)$ such that $\mathcal{A} \models \underline{D}(\mathsf{a}, \mathsf{b})$ and $\mathcal{A} \models \underline{D}'(\mathsf{a}', \mathsf{b}')$.

(b) For every pair $\langle s,t\rangle \in \mathbb{Z}_r^2$ there are pairs $\langle D, D'\rangle \in \mathcal{V}$, $\langle \mathsf{a},\mathsf{b}\rangle = \rho(s,t)$, and $\langle \mathsf{a}',\mathsf{b}'\rangle = \rho(s,t+1)$ such that $\mathcal{A} \models \underline{D}(\mathsf{a},\mathsf{b})$ and $\mathcal{A} \models \underline{D}'(\mathsf{a}',\mathsf{b}')$.

Consequently, the mapping τ constitutes a proper tiling of $\mathbb{Z}_r^2$. □

5.3.3 Replacing the Equality Predicate

Since SF can express reflexivity, symmetry, transitivity, and compatibility with predicates, it is easy to formulate an SF sentence without equality that is equisatisfiable to $\psi_1 \wedge \ldots \wedge \psi_{16} \wedge \chi_1 \wedge \chi_2 \wedge \chi_3 \wedge \eta_1 \wedge \ldots \wedge \eta_{18}$ and uses atoms $E(s,t)$ instead of $s \approx t$ for any terms s,t. In addition to replacing equational atoms as indicated, we conjoin the necessary axioms concerning the fresh predicate symbol E. More precisely, we add the equivalence axioms

$$\begin{aligned}
\psi_1' &:= \forall j.\ E(j,j)\ ,\\
\psi_2' &:= \forall jj'.\ E(j,j') \ \rightarrow\ E(j',j)\ ,\\
\psi_3' &:= \forall jj'j''.\ E(j,j') \wedge E(j',j'') \ \rightarrow\ E(j,j'')\ ,
\end{aligned}$$

and congruence axioms, such as

$$\begin{aligned}
\psi_4' := \forall jj'ii'.\ \bigwedge_{\ell=0}^{\kappa} \Big(&\big(E(j,j') \wedge E(i,i') \wedge J(\underline{\ell},j,i,0) \ \rightarrow\ J(\underline{\ell},j',i',0)\big)\\
&\wedge\ \big(E(j,j) \wedge E(i,i') \wedge J(\underline{\ell},j,i,1) \ \rightarrow\ J(\underline{\ell},j',i',1)\big)\Big)
\end{aligned}$$

and

$$\psi_5' := \forall xyx'y'.\ \bigwedge_{D\in\mathcal{D}} \Big(E(x,x') \wedge E(y,y') \wedge \underline{D}(x,y) \ \rightarrow\ \underline{D}(x',y')\Big)\ .$$

Overall, the additional formulas have a length that lies in $\mathcal{O}\big(\kappa \log \kappa + |\mathcal{D}| \log |\mathcal{D}|\big)$.

All in all, the encoding of domino problems on large tori that we have devised for SF with equality can also be done in SF without equality. Moreover, notice that all the additional subformulas can be transformed into equivalent Horn sentences.

Chapter 6

Interpolation

In the present chapter we show that several of the decidable first-order fragments introduced in Chapter 3 are closed under interpolation. Craig's interpolation theorem [Cra57a, Cra57b] is an important result that has found numerous applications. It is treated in several standard textbooks both from a proof-theoretic perspective, e.g. [TS96], and a model-theoretic point of view, e.g. [CK90]. Moreover, Craig's interpolation theorem is often treated together with *Beth's definability theorem* and *Robinson's joint consistency theorem*, as the latter two can be elegantly proven using the former, see, e.g. [CK90, Fit96, BBJ02]. Historical notes about the early development of interpolation for first-order logic can be found in [TS96], Section 4.6.3. The PhD thesis of Hoogland provides a comprehensive model-theoretically minded introduction to interpolation, see Section 2.3 in [Hoo01]. A recent survey regarding systems for interpolant extraction in the context of verification can be found in [BJ15a].

Proposition 6.0.1 (Craig's interpolation theorem [Cra57a, Cra57b])**.** *Let φ and ψ be two first-order sentences. If $\varphi \models \psi$, then there exists a first-order sentence χ such that*

(i) $\varphi \models \chi$ and $\chi \models \psi$, and

(ii) any predicate symbol, function symbol, or constant symbol that occurs in χ also occurs in φ and in ψ.

The sentence χ is called a Craig interpolant of φ and ψ.

Constructive proofs of Craig's interpolation theorem have been given based on several kinds of proof systems, most prominently sequent calculi, see, e.g. [BL11], Section 8.2, or [Smu95], Chapter XV. A constructive proof based on semantic tableaux is given in [Fit96], Section 8.12. An early approach towards a practically useful method to extract interpolants from resolution and paramodulation refutations is due to Huang [Hua95]. More recent methods intended for practical use can be found in [BJ15b, KV17].

Craig's interpolation theorem has been extended and refined in various ways. For instance, there are variants that say more about the syntactic structure of interpolants than just proclaiming the presence or absence of predicate or function symbols. Two such results are due to Lyndon [Lyn59] and Schulte-Mönting [SM75]. We shall treat the former in more detail below. The result by Schulte-Mönting stipulates that terms t only occur in an interpolant if t corresponds to sufficiently similar terms s_1 and s_2 that occur in the interpolated formulas. The following example is given in [SM75] where C denotes an interpolant of two sentences A and B, i.e. we have $A \models B$ and $A \models C$ and $C \models B$:

> "Let f be a unary function symbol and let $g(\lambda)$, $h(\lambda')$ be terms starting with different function symbols g, h. If f occurs in A only in the term $f(g(\lambda))$ and in B only in the term $f(h(\lambda'))$ then f can occur in C only in terms $f(y)$ where y is a bound variable."
>
> [SM75], page 159.

In contrast to Craig's theorem, Lyndon's interpolation theorem holds only for relational sentences. Consider any formula φ. We treat subformulas $\varphi_1 \to \varphi_2$ of φ as abbreviations for $\neg\varphi_1 \vee \varphi_2$ and subformulas $\varphi_1 \leftrightarrow \varphi_2$ are treated as abbreviations for $(\neg\varphi_1 \vee \varphi_2) \wedge (\neg\varphi_2 \vee \varphi_1)$. We say that a predicate symbol P *occurs positively in* φ if there is an occurrence of some atom $P(\ldots)$ in φ that lies within the scope of an even number of negation signs. Analogously, we say that a predicate symbol P *occurs negatively in* φ if there is an occurrence of some atom $P(\ldots)$ in φ that lies within the scope of an odd number of negation signs.

Proposition 6.0.2 (Lyndon's interpolation theorem [Lyn59])**.** *Let φ and ψ be two relational first-order sentences. If $\varphi \models \psi$, then there exists a relational first-order sentence χ, called a* Craig–Lyndon interpolant of φ and ψ, *such that*

(i) $\varphi \models \chi$ and $\chi \models \psi$, and

(ii) any predicate symbol P occurs positively (negatively) in χ only if it occurs positively (negatively) in φ and in ψ.

Otto [Ott00] strengthened Lyndon's theorem for the case that the interpolated formulas are restricted to quantification that is relativized with unary predicates. More precisely, given some set U of unary predicate symbols, a quantified (sub)formula is U-relativized if it is of the form $\forall x.\, P(x) \to \varphi'$ or $\exists y.\, P(y) \wedge \varphi'$ for some $P \in U$. Then, any two formulas in which all quantified subformulas are U-relativized have a Craig–Lyndon interpolant in which every quantified subformula is U-relativized as well. Otto emphasizes the multi-sorted reading of this interpolation result, when we regard the unary predicates in U as sort predicates.

Viewed from a different angle, the theorem also entails that the class of U-relativized first-order formulas is closed under Craig–Lyndon interpolation. In this context, the theorem could be considered as a first step towards showing closedness of GF under Craig–Lyndon interpolation. However, this direction has been proven to (almost) be a dead end. It is known that there are pairs of GF sentences that satisfy the requirements of Craig's theorem but do not have an interpolant from GF [HMO99, HM02]. The same has been shown for LGF [HM99, HM02].

However, if the requirements towards interpolants are slightly weakened, closedness of the class of GF sentences with respect to the weaker interpolation property can be recovered. To neatly formulate the result, one needs to distinguish between occurrences of predicate symbols in guards and in positions that are not part of guards (*non-guard predicate symbols*).

> "[T]he guards in the interpolant need not be in the common language but they do occur as a guard in either φ or ψ. [...] [T]he non-guards in the interpolant are only required to occur in φ and ψ: not necessarily as non-guards." [HM02], page 389.

By Lemma 3.10.5 and its variant for SGF, this has immediate consequences for SGF and SLGF.

Proposition 6.0.3. *SGF and SLGF are not closed under Craig interpolation.*

On the other, both fragments are closed under the weaker form of interpolation mentioned above.

Regarding closedness under interpolation, there is good news for GNFO: the class of GNFO sentences is closed under Craig interpolation [BBtC13]. In [BtCV16] effective methods for deriving interpolants for GNFO are given: Craig–Lyndon-style interpolation and also relativized interpolation [Ott00]. By Lemma 3.11.4, this entails closedness under interpolation for SGNFO.

Proposition 6.0.4. *The class of SGNFO sentences is closed under Craig–Lyndon interpolation.*

In the following two sections, we intend to show that the Bernays–Schönfinkel fragment and the Ackermann fragment are both closed under Craig–Lyndon interpolation. As in the case of SGNFO, it follows that SF and GBSR without equality, and GAF have the same interpolation property. Our methods will be of a proof-theoretical nature.

6.1 Interpolation for SF and GBSR

In the present, we argue that, given two SF or GBSR sentences φ and ψ without equality such that $\varphi \models \psi$, there is a Craig–Lyndon interpolant χ with $\varphi \models \chi$ and $\chi \models \psi$ that belongs to the Bernays–Schönfinkel fragment (BS). As both SF and GBSR contain BS, the interpolant χ belongs to SF and GBSR as well.

Theorem 6.1.1 (Craig–Lyndon interpolation for GBSR and SF)**.** *Let φ and ψ be GBSR or SF sentences without equality. If $\varphi \models \psi$, then there exists a BS sentence χ (without equality) such that*

(i) $\varphi \models \chi$ and $\chi \models \psi$, and

(ii) any predicate symbol P occurs positively (negatively) in χ only if it occurs positively (negatively) in φ and in ψ.

In order to proof this result, we use the fact that SF and GBSR sentences without equality can be translated into equivalent BS sentences (cf. Lemmas 3.2.5 and 3.5.2) and thus reduce the problem to interpolation in BS. Lemma 6.1.9 states that BS is indeed closed under Craig–Lyndon interpolation. The proof describes how interpolants can be constructed using Bachmair and Ganzinger's *ordered resolution with selection* [BG01] as a proof system. But before we start proving the lemma, we briefly present the used calculus and the required technical notions.

Definition 6.1.2 (Lexicographic path ordering, taken over from [BN98])**.** *Let Σ be a finite vocabulary and let $\succ$ be a total strict order on the symbols in Σ, called* Σ-precedence*. The* lexicographic path ordering (LPO) $\succ_{\text{lpo}}$ *on the set of terms over the vocabulary Σ and* Var *induced by $\succ$ is defined as follows. For any terms s, t we have $s \succ_{\text{lpo}} t$ if and only if*

(1) $t \in \text{vars}(s)$ and $s \neq t$, or

(2) $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$ with $m, n \geq 0$, and

(2.1) there exists i, $1 \leq i \leq m$, with $s_i \succ_{\text{lpo}} t$, or

(2.2) $f \succ g$ and $s \succ_{\text{lpo}} t_j$ for all j, $1 \leq j \leq n$, or

(2.3) $f = g$, $s \succ t_j$ for all j, $1 \leq j \leq n$, and there exists some i, $1 \leq i \leq m$, such that $s_1 = t_1, \ldots, s_{i-1} = t_{i-1}$ and $s_i \succ_{\text{lpo}} t_i$.

Proposition 6.1.3. *For any Σ-precedence $\succ$, the induced LPO $\succ_{\text{lpo}}$ satisfies the following properties.*

(i) $\succ_{\text{lpo}}$ is well-founded*, i.e. all chains $t_1 \succ_{\text{lpo}} t_2 \succ_{\text{lpo}} \ldots$ are finite.*

(ii) For all terms t, t', every m-ary function symbol f, all terms $s_1, \ldots, s_m$, and every i, $1 \leq i \leq m$, we have that $t \succ_{\text{lpo}} t'$ entails

$$f(s_1, \ldots, s_{i-1}, t, s_{i+1}, \ldots, s_m) \succ_{\text{lpo}} f(s_1, \ldots, s_{i-1}, t', s_{i+1}, \ldots, s_m) \ .$$

(iii) For all terms t_1, t_2 and every substitution σ we have that $t_1 \succ_{\text{lpo}} t_2$ entails $t_1\sigma \succ_{\text{lpo}} t_2\sigma$.

(iv) For every term t and every term t' that is proper a subterm of t we have $t \succ_{\text{lpo}} t'$.

Proof. See Theorems 5.4.8 and 5.4.14 in [BN98]. □

On the set of ground terms, i.e. terms without variables, any $\succ_{\text{lpo}}$ constitutes a total ordering. We lift this total ordering on ground terms to a partial ordering on non-ground terms by stipulating $s \succ_{\text{lpo}} t$ if and only if for every substitution σ for which $s\sigma$ and $t\sigma$ are ground we have $s\sigma \succ_{\text{lpo}} t\sigma$. Moreover, we extend $\succ_{\text{lpo}}$ to atoms over the vocabulary Σ and Var by treating predicate symbols like function symbols and atoms like terms. Finally, literals are ordered such that $P(\ldots) \succ_{\text{lpo}} \neg R(\ldots) \succ_{\text{lpo}} R(\ldots)$ whenever the precedence $\succ$ says $P \succ R$.

Definition 6.1.4 (unifier, mgu)**.** *A substitution σ is* at least as general as *a substitution σ' if there exists some substitution τ such that for all variables x we have $(x\sigma)\tau = x\sigma'$. Given two terms s, t, a* unifier of s and t *is a substitution σ for which $s\sigma = t\sigma$. A unifier θ is a* most general unifier (mgu) *of s and t, if θ is a unifier of s and t and if θ is at least as general as any other unifier for s and t. Two terms s and t are called* unifiable*, if there exists a unifier for s and t.*

Proposition 6.1.5. *If two terms are unifiable, then there exists an mgu θ for s and t.*

Proof. By Theorem 4.5.8 in [BN98]. □

Definition 6.1.6 (Ordered resolution with selection, based on $O_S^{\succ}$ [BG01])**.** *Let $\succ_{\text{lpo}}$ be an LPO and let* sel *be a selection function for literals, i.e. given any clause C,* $\text{sel}(C)$ *is a subset of negative literals from C. The following two inference rules constitute a variant of the ordered resolution calculus with selection $O_S^{\succ}$, due to Bachmair and Ganzinger ([BG01], Section 4.3). We use A, B to denote first-order atoms and C, D to denote first-order clauses. The clauses above the vertical lines are the* premises *and the clauses below the line are the* conclusion *of the respective rule.*

$$\frac{C \vee A \qquad \neg B \vee D}{C\theta \vee D\theta} \textit{ binary resolution}$$

where (a) θ is an mgu of A and B, (b) $A\theta$ is strictly maximal in $C\theta \vee A\theta$, i.e. for every literal L in C we have $A\theta \succ_{\text{lpo}} L\theta$, (c) nothing is selected in $C \vee A$, i.e. $\text{sel}(C \vee A) = \emptyset$*, and (d) either $\neg B$ is selected in $\neg B \vee D$, i.e. $\neg B \in \text{sel}(\neg B \vee D)$, or nothing is selected in $\neg B \vee D$ and $\neg B\theta$ is maximal in $\neg B\theta \vee D\theta$, i.e. for every literal L in D we have $\neg B\theta \succeq_{\text{lpo}} L\theta$.*

$$\frac{C \vee A \vee B}{C\theta \vee A\theta} \textit{ positive factorization}$$

where (a) θ is an mgu of A and B, (b) $A\theta$ is maximal in $C\theta \vee A\theta \vee B\theta$, and (c) nothing is selected in $C \vee A \vee B$.

In the context of ordered resolution, we treat clauses as multisets. In particular, the order in which literals occur in clauses does not play a role. Hence, the notation $C \vee L$ does not *mean that the literal L occurs as the right disjunct in a disjunction. It merely denotes a clause in which L may occur in any position, but must occur at least once.*

The calculus $O_S^{\succ}$ from [BG01] has recently been formalized with the help of the Isabelle proof assistant [SBTW18].

Proposition 6.1.7 (Soundness of $O_S^{\succ}$, [BG01, SBTW18])**.** *Let C, D, E be variable-disjoint clauses and let $\bar{\text{x}} := \text{vars}(C)$, $\bar{\text{y}} := \text{vars}(D)$, and $\bar{\text{z}} := \text{vars}(E)$.*

(i) Suppose that E is the result of applying the binary resolution rule from Definition 6.1.6 using the clauses C, D as premises. Then, we have $(\forall \bar{\text{x}}.\, C) \wedge (\forall \bar{\text{y}}.\, D) \models (\forall \bar{\text{z}}.\, E)$.

(ii) Suppose that D is the result of applying the positive factorization rule from Definition 6.1.6 using the clause C as premise. Then, we have $(\forall \bar{\text{x}}.\, C) \models (\forall \bar{\text{z}}.\, D)$.

saturated clause set

We call a clause set N *saturated* with respect to a given term ordering $\succ_{\text{lpo}}$ and a given selection function sel if any application of the rules from Definition 6.1.6 to any clauses from N results in a clause C that is an instance of some clause D in N.

Proposition 6.1.8 (Refutational completeness of $O_S^{\succ}$, [BG01, SBTW18])**.** *Let $\succ_{\text{lpo}}$ be an LPO and let* sel *be a selection function. Let $\square$ denote the empty clause, which can, in addition, be understood as* logical falsity*. Consider a clause set N that is saturated with respect to $\succ_{\text{lpo}}$ and* sel*. We have $\square \in N$ if and only if N is unsatisfiable, i.e. $N \models \texttt{false}$.*

We now have the necessary notions and results at hand to show that BS is closed under Craig–Lyndon interpolation. The general idea of using ordered resolution to prove Craig's interpolation theorem goes back to Harald Ganzinger. In his lecture notes "Logic in Computer Science" (summer term 2002), he outlined the idea for the case of propositional logic. The proof of the following theorem was inspired by Ganzinger's proof sketch. In addition to the ordering constraints, we will make use of selection to achieve a Lyndon-style interpolation property. Additional care has to be taken in order to control the quantifier prefix of interpolants. This applies even more to the closely related proof of Lemma 6.2.5 in the subsequent section.

Lemma 6.1.9. *Let φ and ψ be relational BS sentences (without equality). If $\varphi \models \psi$, then there exists a relational BS sentence χ such that*

(i) $\varphi \models \chi$ and $\chi \models \psi$, and

(ii) any predicate symbol P occurs positively (negatively) in χ only if it occurs positively (negatively) in φ and in ψ.

Proof sketch. In the degenerate cases where φ is unsatisfiable, i.e. $\varphi \models \mathtt{false}$, or where ψ is a tautology, i.e. $\mathtt{true} \models \psi$, we set $\chi := \mathtt{false}$ and $\chi := \mathtt{true}$, respectively. In all other cases we proceed as follows.

Let φ' and ψ' be quantifier-free formulas and let $\bar{\mathrm{u}}, \bar{\mathrm{v}}, \bar{\mathrm{x}}, \bar{\mathrm{y}}$ be tuples of first-order variables such that $\varphi = \exists\bar{\mathrm{y}}\forall\bar{\mathrm{x}}.\,\varphi'$ and $\psi = \exists\bar{\mathrm{v}}\forall\bar{\mathrm{u}}.\,\psi'$. Without loss of generality, we assume that $\bar{\mathrm{u}}, \bar{\mathrm{v}}, \bar{\mathrm{x}}, \bar{\mathrm{y}}$ are pairwise disjoint and that $\varphi' := \bigwedge_i \varphi_i$ and $\psi' := \bigwedge_j \psi_j$ are in conjunctive normal form. φ', ψ'

Let Π_1 be the set of all predicate symbols that occur in φ' but not in ψ', let Π_2 be the set of all predicate symbols that occur positively in φ' but not positively in ψ' and that do not belong to Π_1, let Π_3 be the set of all predicate symbols that occur negatively in φ' but not negatively in ψ' and that do not belong to $\Pi_1 \cup \Pi_2$, let Π_4 be the set of all predicate symbols that occur in φ' and in ψ' but do not belong to $\Pi_1 \cup \Pi_2 \cup \Pi_3$. We construct the formulas $\widehat{\varphi}'$ and $\widehat{\psi}'$ from φ' and ψ', respectively, by simultaneously replacing every literal $\neg P(\bar{s})$ with $P(\bar{s})$ and every literal $P(\bar{s})$ with $\neg P(\bar{s})$ for every $P \in \Pi_2$. Hence, every $P \in \Pi_2$ occurs negatively in $\widehat{\varphi}'$ but not negatively in $\widehat{\psi}'$, and there are no predicate symbols that occur positively in $\widehat{\varphi}'$ but only negatively in $\widehat{\psi}'$. Moreover, we observe that the above transformation preserves (un)satisfiability of φ, $\neg\psi$, and $\varphi \wedge \neg\psi$. More precisely, we have Π_i $\widehat{\varphi}'$, $\widehat{\psi}'$

$\exists\bar{\mathrm{y}}\forall\bar{\mathrm{x}}.\,\varphi' \models \mathtt{false}$ if and only if $\exists\bar{\mathrm{y}}\forall\bar{\mathrm{x}}.\,\widehat{\varphi}' \models \mathtt{false}$,

$\neg\exists\bar{\mathrm{v}}\forall\bar{\mathrm{u}}.\,\psi' \models \mathtt{false}$ if and only if $\neg\exists\bar{\mathrm{v}}\forall\bar{\mathrm{u}}.\,\widehat{\psi}' \models \mathtt{false}$, and

$\big(\exists\bar{\mathrm{y}}\forall\bar{\mathrm{x}}.\,\varphi'\big) \wedge \neg\big(\exists\bar{\mathrm{v}}\forall\bar{\mathrm{u}}.\,\psi'\big) \models \mathtt{false}$ if and only if $\big(\exists\bar{\mathrm{y}}\forall\bar{\mathrm{x}}.\,\widehat{\varphi}'\big) \wedge \neg\big(\exists\bar{\mathrm{v}}\forall\bar{\mathrm{u}}.\,\widehat{\psi}'\big) \models \mathtt{false}$.

Let $\widehat{\varphi}_{\mathrm{Sk}} := \forall\bar{\mathrm{x}}.\,\widehat{\varphi}'\big[y_1/c_1, \ldots, y_{|\bar{\mathrm{y}}|}/c_{|\bar{\mathrm{y}}|}\big]$ where the c_i are fresh Skolem constants. Moreover, let $\widehat{\psi}_{\mathrm{Sk}} := \forall\bar{\mathrm{v}}.\,\neg\widehat{\psi}'\big[u_1/f_1(\bar{\mathrm{v}}), \ldots, u_{|\bar{\mathrm{u}}|}/f_{|\bar{\mathrm{u}}|}(\bar{\mathrm{v}})\big]$ where the f_i are fresh Skolem functions of arity $|\bar{\mathrm{v}}|$. Hence, $\widehat{\varphi}_{\mathrm{Sk}} \wedge \widehat{\psi}_{\mathrm{Sk}}$ is a Skolemized variant of $\big(\exists\bar{\mathrm{y}}\,\forall\bar{\mathrm{x}}.\,\widehat{\varphi}'\big) \wedge \big(\forall\bar{\mathrm{v}}\,\exists\bar{\mathrm{u}}.\,\neg\widehat{\psi}'\big)$, which is semantically equivalent to $\big(\exists\bar{\mathrm{y}}\,\forall\bar{\mathrm{x}}.\,\widehat{\varphi}'\big) \wedge \neg\big(\exists\bar{\mathrm{v}}\,\forall\bar{\mathrm{u}}.\,\widehat{\psi}'\big)$. Therefore, we observe $\widehat{\varphi}_{\mathrm{Sk}}$, $\widehat{\psi}_{\mathrm{Sk}}$

$$\begin{aligned} \varphi \wedge \neg\psi \models \mathtt{false} \quad &\text{if and only if} \quad \big(\exists\bar{\mathrm{y}}\,\forall\bar{\mathrm{x}}.\,\widehat{\varphi}'\big) \wedge \neg\big(\exists\bar{\mathrm{v}}\,\forall\bar{\mathrm{u}}.\,\widehat{\psi}'\big) \models \mathtt{false} \\ &\text{if and only if} \quad \widehat{\varphi}_{\mathrm{Sk}} \wedge \widehat{\psi}_{\mathrm{Sk}} \models \mathtt{false}\,. \end{aligned}$$

Let N be a clause set corresponding to $\widehat{\varphi}_{\mathrm{Sk}}$ such that every P occurring positively (negatively) in N also occurs positively (negatively) in $\widehat{\varphi}_{\mathrm{Sk}}$ — we define N to be the set containing all the implicitly universally quantified clauses $\widehat{\varphi}_i$ from $\widehat{\varphi}_{\mathrm{Sk}}$ whose variables are renamed so that the clauses in N are pairwise variable disjoint. Analogously, let M be the clause set corresponding to $\widehat{\psi}_{\mathrm{Sk}}$ such that every P occurring positively (negatively) in M occurs positively (negatively) in $\widehat{\psi}_{\mathrm{Sk}}$ and, hence, negatively (positively) in $\widehat{\psi}'$. N M

We exhaustively apply ordered resolution with selection to N until the clause set is saturated and call the result N_*. As underlying term ordering we apply an LPO satisfying the following conditions. For all ground literals $P(\ldots)$, $R(\ldots)$, and $\neg R(\ldots)$ we require $P(\ldots) \succ \neg R(\ldots) \succ R(\ldots)$ whenever $P \in \Pi_1$ and $R \notin \Pi_1$. In order to achieve this, we use a precedence $\succ$ for which $P \succ R \succ f \succ c$ for every $P \in \Pi_1$, $R \notin \Pi_1$, every Skolem function f occurring in $\widehat{\psi}_{\mathrm{Sk}}$, and every Skolem constant c occurring in $\widehat{\varphi}_{\mathrm{Sk}}$. We lift the resulting total ordering on ground terms to a (partial) ordering on non-ground terms, atoms, and literals as described below Proposition 6.1.3 on page 169. The selection function sel that we use shall select exactly the literals $\neg P(\bar{s}\,)$ with $P \in \Pi_2 \cup \Pi_3$ in clauses that contain such literals. In all other clauses nothing shall be selected. Let M_* be the result of saturating M in the same way as we have saturated N to obtain N_*.

Note that N_* may be infinite, but may only contain clauses whose literals are instances of the literals in N where variables are either instantiated with variables or with constant symbols c_i. Since φ (and thus also $\widehat{\varphi}_{\mathrm{Sk}}$) is satisfiable and since ordered resolution with selection is sound, N_* does not contain the empty clause. The set M_* may also be infinite. Due to our assumption that ψ is not valid, $\neg\psi$ (and thus also $\widehat{\psi}_{\mathrm{Sk}}$) must be satisfiable. Hence, M_* does not contain the empty clause either.

As our assumption $\varphi \models \psi$ is equivalent to $\varphi \wedge \neg\psi \models \mathtt{false}$ and to $\widehat{\varphi}_{\mathrm{Sk}} \wedge \widehat{\psi}_{\mathrm{Sk}} \models \mathtt{false}$, refutational completeness of ordered resolution with selection entails that there is a (finite) derivation $\mathfrak{D}$ of the empty clause $\Box$ (which at the same stands for *logical falsity*) from the unsatisfiable set of clauses $N_* \cup M_*$. We assume that $\mathfrak{D}$ is based on the same calculus and the same term ordering that we have used to saturate N_* and M_*. Let N'_* be the set of clauses from N_* whose instances are used as premises in this derivation. Since N_* and M_* are both saturated and neither of them contains the empty clause, $\mathfrak{D}$ must indeed make use of clauses from N_*, and, hence, N'_* is not empty. Since N'_* is finite, we can define the sentence $\widehat{\chi}_{\mathrm{Sk}} := \forall \bar{\mathrm{z}}.\, \bigwedge_{C \in N'_*} C$, where we set $\bar{\mathrm{z}} := \mathrm{vars}(N'_*)$. We observe the following properties for $\widehat{\chi}_{\mathrm{Sk}}$ and the underlying clause set N'_*:

(1) $\widehat{\varphi}_{\mathrm{Sk}} \models \widehat{\chi}_{\mathrm{Sk}}$,

(2) $\widehat{\chi}_{\mathrm{Sk}} \wedge \widehat{\psi}_{\mathrm{Sk}} \models \mathtt{false}$,

(3) for every $C \in N'_*$ we have

(3.1) for every literal $P(s_1, \ldots, s_m)$ in C there is a clause $D \in M$ that contains some literal $\neg P(t_1, \ldots, t_m)$, and

(3.2) for every literal $\neg P(s_1, \ldots, s_m)$ in C there is a clause $D \in M$ that contains some literal $P(t_1, \ldots, t_m)$.

Ad (1) and (2). Both observations follow by soundness of ordered resolution with selection. $\Diamond$

Ad (3). Since N_* and M_* are both saturated and do not contain the empty clause, any inference step in $\mathfrak{D}$ that starts from two leaves of the derivation tree involves one clause taken from N'_* and one clause taken from M_*. Consider any such resolution step between clauses $C \in N'_*$ and $D \in M_*$. By case distinction on the possible resolution steps we show that C cannot contain any literal $[\neg]P(\bar{s}\,)$ with $P \in \Pi_1 \cup \Pi_2 \cup \Pi_3$.

Suppose there is a binary resolution step between two clauses $C = C' \vee R(\bar{t}\,) \in N'_*$ and $D = D' \vee \neg R(\bar{t}'\,) \in M_*$ over the literals $R(\bar{t}\,)$ and $\neg R(\bar{t}'\,)$ such that C contains some literal $[\neg]P(\bar{s}\,)$ with $P \in \Pi_1$. Since R occurs in N_* and in M_*, we have $R \notin \Pi_1$. Hence, we get $P(\bar{s}\,) \succ R(\bar{t}\,)$. Due to the order restrictions in ordered resolution, $R(\bar{t}\,)\tau$ must be maximal in $C\tau$, where τ is the unifier that is used in the resolution step to unify $R(\bar{t}\,)$ and $R(\bar{t}'\,)$. But this contradicts $P(\bar{s}\,) \succ R(\bar{t}\,)$, as the latter entails $P(\bar{s}\,)\tau \succ R(\bar{t}\,)\tau$.

Suppose there is a binary resolution step between two clauses $C = C' \vee \neg R(\bar{t}\,) \in N'_*$ and $D = D' \vee R(\bar{t}'\,) \in M_*$ over the literals $\neg R(\bar{t}\,)$ and $R(\bar{t}'\,)$ such that C contains some literal $[\neg]P(\bar{s}\,)$ with $P \in \Pi_1$. Since R occurs negatively in N_* and positively in M_*, we conclude $R \notin \Pi_1 \cup \Pi_2 \cup \Pi_3$. Hence, we have that $P(\bar{s}\,) \succ R(\bar{t}\,)$, which entails $P(\bar{s}\,) \succ \neg R(\bar{t}\,)$, and

$\neg R(\bar{t}\,)$ is not selected in C. But then, due to the order restrictions in ordered resolution, $\neg R(\bar{t}\,)\tau$ must be maximal in $C\tau$, where τ is the unifier that is used to unify $R(\bar{t}\,)$ and $R(\bar{t}')$. But this contradicts $P(\bar{s}\,) \succ \neg R(\bar{t}\,)$, as the latter entails $P(\bar{s}\,)\tau \succ \neg R(\bar{t}\,)\tau$.

Suppose there is a binary resolution step between two clauses $C = C' \vee R(\bar{t}\,) \in N'_*$ and $D = D' \vee \neg R(\bar{t}') \in M_*$ over the literals $R(\bar{t}\,)$ and $\neg R(\bar{t}')$ such that C contains some literal $\neg P(\bar{s}\,)$ with $P \in \Pi_2 \cup \Pi_3$. Since $\neg P(\bar{s}\,)$ is selected in C by sel, this resolution step is not admitted.

Suppose there is an binary resolution step between two clauses $C = C' \vee \neg R(\bar{t}\,) \in N'_*$ and $D = D' \vee R(\bar{t}') \in M_*$ over the literals $\neg R(\bar{t}\,)$ and $R(\bar{t}')$ such that C contains some literal $\neg P(\bar{s}\,)$ with $P \in \Pi_2 \cup \Pi_3$. Since R occurs negatively in N_* and positively in M_*, it must occur negatively in $\widehat{\psi}'$, and thus $R \notin \Pi_1 \cup \Pi_2 \cup \Pi_3$. Hence, the literal $\neg R(\bar{t}\,)$ is not selected in C. Since, on the other hand, there is a selected literal in C by sel, namely $\neg P(\bar{s}\,)$, this resolution step is not admitted.

Consequently, the result of any inference step starting from two leaf nodes of the derivation tree of $\mathfrak{D}$ cannot contain any predicate symbol $P \in \Pi_1$ and it cannot contain any literal $\neg R(\ldots)$ with $R \in \Pi_2 \cup \Pi_3$.

By an inductive argument (over the height of derivation trees), this leads to the observation that none of the clauses from N_* that are involved in the derivation $\mathfrak{D}$ can contain any predicate symbols from Π_1 or any negative literals $\neg R(\ldots)$ with $R \in \Pi_2 \cup \Pi_3$. Since N'_* contains only clauses that are involved in $\mathfrak{D}$, Condition (3.2) is satisfied. By construction of N_* from $\varphi = \exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \varphi'$ via $\exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \widehat{\varphi}'$ and $\widehat{\varphi}_{\mathrm{Sk}}$, Condition (3.1) is satisfied as well. ◊

Since $\widehat{\chi}_{\mathrm{Sk}}$ contains exclusively constant symbols c_i, we can easily construct $\widehat{\chi}'$ from $\widehat{\chi}_{\mathrm{Sk}}$'s matrix by de-Skolemization, i.e. $\widehat{\chi}_{\mathrm{Sk}} = \forall \bar{\mathrm{z}}.\, \widehat{\chi}'\big[y_1/c_1, \ldots, y_{|\bar{\mathrm{y}}|}/c_{|\bar{\mathrm{y}}|}\big]$. Furthermore, we construct the formula χ' from $\widehat{\chi}'$ by simultaneously replacing every literal $\neg P(\bar{s}\,)$ by $P(\bar{s}\,)$ and every literal $P(\bar{s}\,)$ by $\neg P(\bar{s}\,)$ for every $P \in \Pi_2$. Finally, we set $\chi := \exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{z}}.\, \chi'$. $\widehat{\chi}'$, χ', χ

It remains to prove the following properties:

(4) $\varphi \models \chi$ and

(5) $\chi \wedge \neg\psi \models \mathtt{false}$.

Ad (4). For every model $\mathcal{A} \models \exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \widehat{\varphi}'$ there is some model $\mathcal{B} \models \widehat{\varphi}_{\mathrm{Sk}}$ such that $\mathcal{A}$ and $\mathcal{B}$ differ only in their interpretation of the Skolem constants $c_1, \ldots, c_{|\bar{\mathrm{y}}|}$. By (1) and because of $\widehat{\chi}_{\mathrm{Sk}} \models \exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \widehat{\chi}'$, we get $\mathcal{B} \models \exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \widehat{\chi}'$. Since $\mathcal{B}$ differs from $\mathcal{A}$ only in the interpretation of symbols that do not occur in $\exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \widehat{\chi}'$, $\mathcal{A}$ is also a model of $\exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \widehat{\chi}'$. Hence, $\exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \widehat{\varphi}' \models \exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \widehat{\chi}'$, which can equivalently be written as $\big(\exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \widehat{\varphi}'\big) \wedge \neg\big(\exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \widehat{\chi}'\big) \models \mathtt{false}$.

Since $\big(\exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \widehat{\varphi}'\big) \wedge \neg\big(\exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \widehat{\chi}'\big) \models \mathtt{false}$ holds if and only if $\big(\exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \varphi'\big) \wedge \neg\big(\exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \chi'\big) \models \mathtt{false}$, and since the latter is equivalent to $\big(\exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \varphi'\big) \models \big(\exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \chi'\big)$ we in the end get $\varphi \models \chi$. ◊

Ad (5). The formula $\widehat{\chi}_{\mathrm{Sk}} \wedge \widehat{\psi}_{\mathrm{Sk}}$ can be conceived as a Skolemized variant of $\big(\exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \widehat{\chi}'\big) \wedge \big(\forall \bar{\mathrm{v}}\, \exists \bar{\mathrm{u}}.\, \neg\widehat{\psi}'\big)$, which is semantically equivalent to $\big(\exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \widehat{\chi}'\big) \wedge \neg\big(\exists \bar{\mathrm{v}}\, \forall \bar{\mathrm{u}}.\, \widehat{\psi}'\big)$. Hence, we have $\widehat{\chi}_{\mathrm{Sk}} \wedge \widehat{\psi}_{\mathrm{Sk}} \models \mathtt{false}$ if and only if $\big(\exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \widehat{\chi}'\big) \wedge \neg\big(\exists \bar{\mathrm{v}}\, \forall \bar{\mathrm{u}}.\, \widehat{\psi}'\big) \models \mathtt{false}$. As we, in addition, observe that $\big(\exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \widehat{\chi}'\big) \wedge \neg\big(\exists \bar{\mathrm{v}}\, \forall \bar{\mathrm{u}}.\, \widehat{\psi}'\big) \models \mathtt{false}$ holds if and only if $\big(\exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \chi'\big) \wedge \neg\big(\exists \bar{\mathrm{v}}\, \forall \bar{\mathrm{u}}.\, \psi'\big) \models \mathtt{false}$, we in the end get

$$\begin{array}{lll} \widehat{\chi}_{\mathrm{Sk}} \wedge \widehat{\psi}_{\mathrm{Sk}} \models \mathtt{false} & \text{if and only if} & \big(\exists \bar{\mathrm{y}}\, \forall \bar{\mathrm{x}}.\, \widehat{\chi}'\big) \wedge \neg\big(\exists \bar{\mathrm{v}}\, \forall \bar{\mathrm{u}}.\, \widehat{\psi}'\big) \models \mathtt{false} \\ & \text{if and only if} & \chi \wedge \neg\psi \models \mathtt{false}\ . \end{array}$$

By (2), this yields $\chi \wedge \neg\psi \models \mathtt{false}$. ◊

Because of the equivalence of $\chi \wedge \neg\psi \models \texttt{false}$ and $\chi \models \psi$, we have shown that χ satisfies Requirement (i) of the lemma.

Due to (3) and due to the way χ is constructed from N'_*, every positive occurrence of a predicate symbol P in χ entails the existence of a negative occurrence of P in $\neg\psi$, and every negative occurrence of a predicate symbol P in χ entails the existence of a positive occurrence of P in $\neg\psi$. Consequently, χ satisfies Requirement (ii) as well. □

6.2 Interpolation for GAF

After having shown that SF and GBSR without equality are closed under Craig–Lyndon interpolation, we now develop the analogous result for GAF. More precisely, we argue that any two GAF sentences φ and ψ with $\varphi \models \psi$ have a Craig–Lyndon interpolant χ with $\varphi \models \chi$ and $\chi \models \psi$ that belongs to the Ackermann fragment (AF). As GAF contains AF, the interpolant χ is also a GAF sentence.

Theorem 6.2.1 (Craig–Lyndon interpolation for GAF)**.** *Let φ and ψ be GAF sentences. If $\varphi \models \psi$, then there exists an AF sentence χ such that*

(i) $\varphi \models \chi$ and $\chi \models \psi$, and

(ii) any predicate symbol P occurs positively (negatively) in χ only if it occurs positively (negatively) in φ and in ψ.

As for the SF and GBSR case, we use the fact that GAF sentences can be translated into equivalent AF sentences (cf. Lemma 3.8.4). This reduces the problem to interpolation in AF. Again, we use the fact that interpolants can be constructed using Bachmair and Ganzinger's ordered resolution with selection (Definition 6.1.6). This time, however, this yields only an intermediate form of interpolants that we cannot immediately de-Skolemize into AF sentences. At this point, we employ known techniques to replace terms with quantified variables (cf. [BL11], Section 8.2), if the terms start with function symbols that do not belong to the common vocabulary of the interpolated formulas. This replacement preserves the logical entailments between the interpolant and the interpolated formulas. As we shall argue about the soundness of the replacement method using a certain sequent calculus, we introduce its derivation rules before we get started proving the interpolation result.

Definition 6.2.2 (Modified sequent calculus LK, adapted from [BL11])**.** *The following set of derivation rules defines a slightly modified variant of the calculus LK, that we shall refer to as* $\mathrm{LK}_{\top\bot}$.[1]

Axioms:

$$\frac{}{A \vdash A} \qquad \frac{}{\vdash \texttt{true}} \qquad \frac{}{\texttt{false} \vdash}$$

where A may be any non-equational first-order atom.

Introduction of $\wedge$:

$$\frac{\varphi, \Gamma \vdash \Delta}{(\varphi \wedge \psi), \Gamma \vdash \Delta} \wedge : l_1 \qquad \frac{\psi, \Gamma \vdash \Delta}{(\varphi \wedge \psi), \Gamma \vdash \Delta} \wedge : l_2 \qquad \frac{\Gamma \vdash \Delta, \varphi \qquad \Gamma \vdash \Delta, \psi}{\Gamma \vdash \Delta, (\varphi \wedge \psi)} \wedge : r$$

[1] The calculus $\mathrm{LK}_{\top\bot}$ from [BL11] is, according to the authors, almost identical to the original LK by Gentzen [Gen35a, Gen35b]. The adaptations to $\mathrm{LK}_{\top\bot}$ [BL11] made in the present thesis are inessential. The rules dedicated to implication are left out — they are derivable using the rules for $\neg$ and $\vee$. We do not syntactically distinguish free from bound first-order variables. Our permutation rule is more compact. And, finally, we have not integrated implicit contraction into the cut rule — this can be simulated by using the contraction rule appropriately prior to cuts.

Introduction of $\vee$*:*

$$\frac{\varphi, \Gamma \vdash \Delta \qquad \psi, \Gamma \vdash \Delta}{(\varphi \vee \psi), \Gamma \vdash \Delta} \vee : l \qquad \frac{\Gamma \vdash \Delta, \varphi}{\Gamma \vdash \Delta, (\varphi \vee \psi)} \vee : r_1 \qquad \frac{\Gamma \vdash \Delta, \psi}{\Gamma \vdash \Delta, (\varphi \vee \psi)} \vee : r_2$$

Introduction of $\neg$*:*

$$\frac{\Gamma \vdash \Delta, \varphi}{\neg\varphi, \Gamma \vdash \Delta} \neg : l \qquad \frac{\varphi, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg\varphi} \neg : r$$

Introduction of $\forall$*:*

$$\frac{\varphi\big[x/t\big], \Gamma \vdash \Delta}{\forall x.\, \varphi, \Gamma \vdash \Delta} \forall : l$$

where t *may be any term whose variables are free in* $\varphi\big[x/t\big]$*.*

$$\frac{\Gamma \vdash \Delta, \varphi}{\Gamma \vdash \Delta, \forall x.\, \varphi} \forall : r$$

where the variable x *may not occur in* Γ, Δ*.*

Introduction of $\exists$*:*

$$\frac{\varphi, \Gamma \vdash \Delta}{\exists y.\, \varphi, \Gamma \vdash \Delta} \exists : l$$

where the variable y *may not occur in* Γ, Δ*.*

$$\frac{\Gamma \vdash \Delta, \varphi\big[y/t\big]}{\Gamma \vdash \Delta, \exists y.\, \varphi} \exists : r$$

where t *may be any term whose variables are free in* $\varphi\big[y/t\big]$*.*

Permutation:

$$\frac{\Gamma \vdash \Delta}{\Gamma' \vdash \Delta'}\, p$$

where the sequence Γ' *is a permutation of* Γ *and the sequence* Δ' *is a permutation of* Δ*.*

Weakening:

$$\frac{\Gamma \vdash \Delta}{\varphi, \Gamma \vdash \Delta}\, w : l \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, \varphi}\, w : r$$

Contraction:

$$\frac{\varphi, \varphi, \Gamma \vdash \Delta}{\varphi, \Gamma \vdash \Delta}\, c : l \qquad \frac{\Gamma \vdash \Delta, \varphi, \varphi}{\Gamma \vdash \Delta, \varphi}\, c : r$$

Cut:

$$\frac{\Gamma \vdash \Delta, \varphi \qquad \varphi, \Gamma' \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\, \mathrm{cut}(\varphi)$$

Proposition 6.2.3 (Soundness and completeness of $\mathrm{LK}_{\top\bot}$ for first-order sentences without equality)**.** *Let* φ, ψ *be two first-order sentences without equality. We have* $\varphi \models \psi$ *if and only if* $\varphi \vdash \psi$ *is derivable in* $\mathrm{LK}_{\top\bot}$*.*

Proposition 6.2.4 (Cut-elimination for $\mathrm{LK}_{\top\bot}$, cf. Theorem 5.2.3 in [BL11])**.** *Let* φ, ψ *be two first-order sentences without equality. If* $\varphi \vdash \psi$ *is derivable in* $\mathrm{LK}_{\top\bot}$*, then it is also derivable using the rule* $\mathrm{cut}(\chi)$ *only for atomic* χ*.*

This gives us the necessary tools at hand to show that AF is closed under Craig–Lyndon interpolation. Compared to the BS case, we have to take extra steps to ensure that the quantifier

prefix of interpolants fits the requirements of AF. This is the point where the proof system $\mathrm{LK}_{\top\bot}$ enters the stage.

Lemma 6.2.5. *Let φ and ψ be AF sentences (without equality) in which the only Boolean connectives are $\wedge, \vee, \neg$. If $\varphi \models \psi$, then there exists a relational AF sentence χ without equality such that*

(i) $\varphi \models \chi$ and $\chi \models \psi$, and

(ii) any predicate symbol P occurs positively (negatively) in χ only if it occurs positively (negatively) in φ and in ψ.

Proof sketch. In the degenerate cases where φ is unsatisfiable, i.e. $\varphi \models \mathtt{false}$, or where ψ is a tautology, i.e. $\mathtt{true} \models \psi$, we set $\chi := \mathtt{false}$ and $\chi := \mathtt{true}$, respectively. In all other cases we proceed as follows.

φ', ψ' Let φ' and ψ' be quantifier-free formulas and let $\bar{\mathrm{u}}, \bar{\mathrm{w}}, \bar{\mathrm{y}}, \bar{\mathrm{z}}$ be tuples of variables such that $\varphi = \exists\bar{\mathrm{u}}\forall v\exists\bar{\mathrm{w}}.\, \varphi'(\bar{\mathrm{u}}, v, \bar{\mathrm{w}})$ and $\psi = \exists\bar{\mathrm{y}}\forall x\exists\bar{\mathrm{z}}.\, \psi'(\bar{\mathrm{y}}, x, \bar{\mathrm{z}})$. Without loss of generality, we assume that $\bar{\mathrm{u}}, \{v\}, \bar{\mathrm{w}}, \bar{\mathrm{y}}, \{x\}, \bar{\mathrm{z}}$ are pairwise disjoint and that $\varphi' := \bigwedge_i \varphi_i$ and $\psi' := \bigwedge_j \psi_j$ are in conjunctive normal form.

Π_i We define the sets $\Pi_1, \Pi_2, \Pi_3, \Pi_4$ like in the proof of Lemma 6.1.9. We construct the formulas

$\widehat{\varphi}'$, $\widehat{\psi}'$ $\widehat{\varphi}'$ and $\widehat{\psi}'$ from φ' and ψ', respectively, by simultaneously replacing every literal $\neg P(\bar{s}\,)$ by $P(\bar{s}\,)$ and every literal $P(\bar{s}\,)$ by $\neg P(\bar{s}\,)$ for every $P \in \Pi_2$. Hence, every $P \in \Pi_2$ occurs negatively in $\widehat{\varphi}'$ but not negatively in $\widehat{\psi}'$, and there are no predicate symbols that occur positively in $\widehat{\varphi}'$ but only negatively in $\widehat{\psi}'$. Moreover, we observe that the above transformation preserves (un)satisfiability of φ, $\neg\psi$, and $\varphi \wedge \neg\psi$. More precisely, we have

$$\exists\bar{\mathrm{u}}\,\forall v\exists\bar{\mathrm{w}}.\, \varphi' \models \mathtt{false} \text{ if and only if } \exists\bar{\mathrm{u}}\,\forall v\exists\bar{\mathrm{w}}.\, \widehat{\varphi}' \models \mathtt{false},$$

$$\neg\exists\bar{\mathrm{y}}\,\forall x\,\exists\bar{\mathrm{z}}.\, \psi' \models \mathtt{false} \text{ if and only if } \neg\exists\bar{\mathrm{y}}\,\forall x\,\exists\bar{\mathrm{z}}.\, \widehat{\psi}' \models \mathtt{false}, \text{ and}$$

$$\big(\exists\bar{\mathrm{u}}\,\forall v\exists\bar{\mathrm{w}}.\, \varphi'\big) \wedge \neg\big(\exists\bar{\mathrm{y}}\,\forall x\,\exists\bar{\mathrm{z}}.\, \psi'\big) \models \mathtt{false} \text{ if and only if } \big(\exists\bar{\mathrm{u}}\,\forall v\exists\bar{\mathrm{w}}.\, \widehat{\varphi}'\big) \wedge \neg\big(\exists\bar{\mathrm{y}}\,\forall x\,\exists\bar{\mathrm{z}}.\, \widehat{\psi}'\big) \models \mathtt{false}.$$

$\widehat{\varphi}_{\mathrm{Sk}}$, $\widehat{\psi}_{\mathrm{Sk}}$ Let $\widehat{\varphi}_{\mathrm{Sk}} := \forall v.\, \widehat{\varphi}'\big[u_1/c_1, \ldots, u_{|\bar{\mathrm{u}}|}/c_{|\bar{\mathrm{u}}|}, w_1/f_1(v), \ldots, w_{|\bar{\mathrm{w}}|}/f_{|\bar{\mathrm{w}}|}(v)\big]$ where the c_i are fresh Skolem constants and the f_i are fresh unary Skolem functions. Moreover, let $\widehat{\psi}_{\mathrm{Sk}} := \forall\bar{\mathrm{y}}\bar{\mathrm{z}}.\, \neg\widehat{\psi}'\big[x/g(\bar{\mathrm{y}})\big]$ where the g is a fresh Skolem function of arity $|\bar{\mathrm{y}}|$. Hence, $\widehat{\varphi}_{\mathrm{Sk}} \wedge \widehat{\psi}_{\mathrm{Sk}}$ is a Skolemized variant of $\big(\exists\bar{\mathrm{u}}\,\forall v\exists\bar{\mathrm{w}}.\, \widehat{\varphi}'\big) \wedge \big(\forall\bar{\mathrm{y}}\,\exists x\,\forall\bar{\mathrm{z}}.\, \neg\widehat{\psi}'\big)$, which is equivalent to $\big(\exists\bar{\mathrm{u}}\,\forall v\exists\bar{\mathrm{w}}.\, \widehat{\varphi}'\big) \wedge \neg\big(\exists\bar{\mathrm{y}}\,\forall x\,\exists\bar{\mathrm{z}}.\, \widehat{\psi}'\big)$. Therefore, we observe

$$\begin{aligned} \varphi \wedge \neg\psi \models \mathtt{false} \quad &\text{if and only if} \quad \big(\exists\bar{\mathrm{u}}\,\forall v\exists\bar{\mathrm{w}}.\, \widehat{\varphi}'\big) \wedge \big(\forall\bar{\mathrm{y}}\,\exists x\,\forall\bar{\mathrm{z}}.\, \neg\widehat{\psi}'\big) \models \mathtt{false} \\ &\text{if and only if} \quad \widehat{\varphi}_{\mathrm{Sk}} \wedge \widehat{\psi}_{\mathrm{Sk}} \models \mathtt{false}\,. \end{aligned}$$

N Let N be a clause set corresponding to $\widehat{\varphi}_{\mathrm{Sk}}$ such that every P occurring positively (negatively) in N also occurs positively (negatively) in $\widehat{\varphi}_{\mathrm{Sk}}$ — we define N to be the set containing all the implicitly universally quantified clauses $\widehat{\varphi}_i$ from $\widehat{\varphi}_{\mathrm{Sk}}$ whose variables are renamed so that the

M clauses in N are pairwise variable disjoint. Analogously, let M be the clause set corresponding to $\widehat{\psi}_{\mathrm{Sk}}$ such that every P occurring positively (negatively) in M occurs positively (negatively) in $\widehat{\psi}_{\mathrm{Sk}}$ and negatively (positively) in $\widehat{\psi}'$.

We exhaustively apply ordered resolution with selection to N until the clause set is saturated

N_* and call the result N_*. As underlying term ordering we apply an ordering defined like in the proof of Lemma 6.1.9 based on an LPO induced by a precedence $\succ$ for which $P \succ R \succ f \succ c$ for any $P \in \Pi_1$, $R \notin \Pi_1$, any Skolem function f occurring in $\widehat{\varphi}_{\mathrm{Sk}}$ or $\widehat{\psi}_{\mathrm{Sk}}$, and any Skolem constant c occurring in $\widehat{\varphi}_{\mathrm{Sk}}$ or $\widehat{\psi}_{\mathrm{Sk}}$. The used selection function sel selects exactly the literals $\neg P(\bar{s}\,)$ with

M_* $P \in \Pi_2 \cup \Pi_3$ in all clauses that contain such literals. Let M_* be the result of saturating M in the same way as we have saturated N to obtain N_*.

Claim I: An *Ackermann-like clause* is a clause that contains at most one variable and only function symbols of arity at most 1. *Ackermann-like clauses*

Consider any two Ackermann-like clauses $C := [\neg]A \vee C'$ and $D := [\neg]B \vee D'$ with unifiable atoms A and B. Let σ be a most general unifier of A and B. Then, $(C' \vee D')\sigma$ is an Ackermann-like clause.

Proof: Since σ is a most general unifier, for every variable x with $\sigma(x) \neq x$ we observe that $\sigma(x)$ is some term that (a) contains at most one variable and (b) that does not contain any function symbols of arity larger than one. But then, applying σ to the Ackermann-like clause $C' \vee D'$ clearly results in an Ackermann-like clause $(C' \vee D')\sigma$. $\Diamond$

Notice that N_* may be infinite. Moreover, by Claim I, N_* contains exclusively Ackermnann-like clauses over the vocabulary underlying N. Since φ (and thus also $\widehat{\varphi}_{\mathrm{Sk}}$) is satisfiable and since ordered resolution with selection is sound, N_* does not contain the empty clause. The set M_* may also be infinite. Due to our assumption that ψ is not valid, $\neg\psi$ (and thus also $\widehat{\psi}_{\mathrm{Sk}}$) must be satisfiable. Hence, M_* does not contain the empty clause either.

As our assumption $\varphi \models \psi$ is equivalent to $\varphi \wedge \neg\psi \models \mathtt{false}$ and to $\widehat{\varphi}_{\mathrm{Sk}} \wedge \widehat{\psi}_{\mathrm{Sk}} \models \mathtt{false}$, refutational completeness of ordered resolution with selection entails that there is a (finite) derivation $\mathfrak{D}$ of the empty clause $\square$ from the unsatisfiable set of clauses $N_* \cup M_*$. We assume that $\mathfrak{D}$ is based on the same calculus and the same term ordering that we have used to saturate N_* and M_*. Let N'_* be the set of clauses from N_* whose instances are used as premises in this derivation. Since N_* and M_* are both saturated and neither of them contains the empty clause, $\mathfrak{D}$ must indeed make use of clauses from N_*, and, hence, N'_* is not empty. Since N'_* is finite, we may define the sentence $\widehat{\chi}_{\mathrm{Sk}} := \forall v.\, \widehat{\chi}'_{\mathrm{Sk}}$ with $\mathfrak{D}$, N'_*, $\widehat{\chi}_{\mathrm{Sk}}$, $\widehat{\chi}'_{\mathrm{Sk}}$

$$\widehat{\chi}'_{\mathrm{Sk}} := \bigwedge_{C(x) \in N'_*} C(v) \ ,$$

where v is a fresh variable not occurring in N_*. Like in the proof of Lemma 6.1.9, we observe the following properties for $\widehat{\chi}_{\mathrm{Sk}}$ and the underlying clause set N'_*:

(1) $\widehat{\varphi}_{\mathrm{Sk}} \models \widehat{\chi}_{\mathrm{Sk}}$,

(2) $\widehat{\chi}_{\mathrm{Sk}} \wedge \widehat{\psi}_{\mathrm{Sk}} \models \mathtt{false}$,

(3) for every $C \in N'_*$ we have

  (3.1) for every literal $P(s_1, \ldots, s_m)$ in C there is a clause $D \in M$ that contains some literal $\neg P(t_1, \ldots, t_m)$, and

  (3.2) for every literal $\neg P(s_1, \ldots, s_m)$ in C there is a clause $D \in M$ that contains some literal $P(t_1, \ldots, t_m)$.

Claim II: From the sentence $\widehat{\chi}_{\mathrm{Sk}}$ we can construct a relational sentence $\widehat{\chi} := \exists \bar{\mathrm{y}}' \forall v \exists \bar{\mathrm{z}}'.\, \widehat{\chi}'$ with quantifier-free $\widehat{\chi}'$ such that $\widehat{\varphi}_{\mathrm{Sk}} \models \widehat{\chi}$ and $\widehat{\chi} \wedge \widehat{\psi}_{\mathrm{Sk}} \models \mathtt{false}$. $\widehat{\chi}$, $\widehat{\chi}'$

Proof sketch: If $\widehat{\chi}_{\mathrm{Sk}}$ is relational, we are done. Suppose it is not. Then, $\widehat{\chi}_{\mathrm{Sk}}$ and $\neg\widehat{\psi}_{\mathrm{Sk}}$ do not share any function symbols. We argue by using a construction due to Baaz and Leitsch (proof of Lemma 8.2.2 in [BL11]). The construction relies on the sequent calculus $\mathrm{LK}_{\top\bot}$ given in Definition 6.2.2. We use a part of the construction here with several adjustments that are required to properly deal with the leading universal quantifier in $\widehat{\chi}_{\mathrm{Sk}}$.

By completeness of $\mathrm{LK}_{\top\bot}$ (Proposition 6.2.3), there is some derivation π_0 of the form π_0, π_1, π_2

$$\dfrac{\widehat{\varphi}_{\mathrm{Sk}} \ \vdash \ \forall v.\, \widehat{\chi}'_{\mathrm{Sk}} \quad \overset{\pi_1}{} \qquad \forall v.\, \widehat{\chi}'_{\mathrm{Sk}} \ \vdash \ \neg\widehat{\psi}_{\mathrm{Sk}} \quad \overset{\pi_2}{}}{\widehat{\varphi}_{\mathrm{Sk}} \ \vdash \ \widehat{\psi}_{\mathrm{Sk}}} \ \mathrm{cut}(\forall v.\, \widehat{\chi}'_{\mathrm{Sk}})$$

where π_1 and π_2 are certain subderivations. By Proposition 6.2.4, we may assume π_1, π_2 to be free of non-atomic cuts.

π_1', π_2' In these subderivations the following subderivations π_1' and π_2' must occur, respectively:

$$\pi_1\text{:}\quad \dfrac{\begin{array}{c}\pi_1''\\ \Gamma_1 \vdash \Delta_1, \widehat{\chi}'_{\mathrm{Sk}}\end{array}}{\Gamma_1 \vdash \Delta_1, \forall v.\, \widehat{\chi}'_{\mathrm{Sk}}}\,\forall{:}r \qquad \pi_2\text{:}\quad \dfrac{\begin{array}{c}\pi_2''\\ \Gamma_2, \widehat{\chi}'_{\mathrm{Sk}}[v/t] \vdash \Delta_2\end{array}}{\Gamma_2, \forall v.\, \widehat{\chi}'_{\mathrm{Sk}} \vdash \Delta_2}\,\forall{:}l$$

We may assume without loss of generality that π_2 contains exactly one subderivation of the form that π_2' exhibits. We can do this, because $\neg\widehat{\psi}_{\mathrm{Sk}}$ and $\widehat{\chi}'_{\mathrm{Sk}}$ do not share any function symbols. If there are more than one such subderivations, then π_2 must merge all the resulting subformulas $\forall v.\, \widehat{\chi}'_{\mathrm{Sk}}$ into one by contraction steps. But then, π_2 can be rewritten so that all of these contraction steps are done before the introduction of $\forall v$ in front of $\widehat{\chi}'_{\mathrm{Sk}}$. Moreover, any cut over atoms containing function symbols from $\widehat{\chi}'_{\mathrm{Sk}}$ can be done before introduction of $\forall v$. Hence, we can in fact assume the following.

Claim III: Γ_2 and Δ_2 in π_2' do not contain any function symbol occurring in $\widehat{\chi}'_{\mathrm{Sk}}$.

We modify π_1 and π_2 in such a way that every occurrence of subderivations of the form π_1'
ρ_1', ρ_2' or π_2' are replaced by the derivations ρ_1' and ρ_2' shown below:

$$\rho_1'\text{:}\quad \dfrac{\begin{array}{c}\vdots\\ \dfrac{\begin{array}{c}\vdots\\ \dfrac{\begin{array}{c}\vdots\\ \dfrac{\begin{array}{c}\pi_1''\\ \Gamma_1 \vdash \Delta_1, \widehat{\chi}'_{\mathrm{Sk}}\end{array}}{\Gamma_1 \vdash \Delta_1, \exists z_1'.\, \widehat{\chi}'_{\mathrm{Sk}}[s_1/z_1']}\,\exists{:}r\\ \vdots\\ \Gamma_1 \vdash \Delta_1, \exists z_{k-1}' \ldots z_1'.\, \widehat{\chi}'_{\mathrm{Sk}}[s_1/z_1'] \ldots [s_{k-1}/z_{k-1}']\end{array}}{\Gamma_1 \vdash \Delta_1, \exists z_k' z_{k-1}' \ldots z_1'.\, \widehat{\chi}'_{\mathrm{Sk}}[s_1/z_1'] \ldots [s_{k-1}/z_{k-1}'][s_k/z_k']}\,\exists{:}r\\ \Gamma_1 \vdash \Delta_1, \forall v \exists z_k' z_{k-1}' \ldots z_1'.\, \widehat{\chi}'_{\mathrm{Sk}}[s_1/z_1'] \ldots [s_{k-1}/z_{k-1}'][s_k/z_k']\end{array}}{\Gamma_1 \vdash \Delta_1, \exists y_1' \forall v \exists z_k' z_{k-1}' \ldots z_1'.\, \widehat{\chi}'_{\mathrm{Sk}}[s_1/z_1'] \ldots [s_{k-1}/z_{k-1}'][s_k/z_k'][t_1/y_1']}\,\exists{:}r\\ \vdots\\ \Gamma_1 \vdash \Delta_1, \exists y_{\ell-1}' \ldots y_1' \forall v \exists z_k' \ldots z_1'.\, \widehat{\chi}'_{\mathrm{Sk}}[s_1/z_1'] \ldots [s_k/z_k'][t_1/y_1'] \ldots [t_{\ell-1}/y_{\ell-1}']\end{array}}{\Gamma_1 \vdash \Delta_1, \exists y_\ell' \ldots y_1' \forall v \exists z_k' \ldots z_1'.\, \widehat{\chi}'_{\mathrm{Sk}}[s_1/z_1'] \ldots [s_k/z_k'][t_1/y_1'] \ldots [t_\ell/y_\ell']}\,\exists{:}r$$

$$\rho_2'\text{:}\quad \dfrac{\begin{array}{c}\vdots\\ \dfrac{\begin{array}{c}\vdots\\ \dfrac{\begin{array}{c}\vdots\\ \dfrac{\begin{array}{c}\pi_2'''\\ \Gamma_2, \widehat{\chi}''_{\mathrm{Sk}}[v/t] \vdash \Delta_2\end{array}}{\Gamma_2, \exists z_1'.\, \widehat{\chi}''_{\mathrm{Sk}}[v/t] \vdash \Delta_2}\,\exists{:}l\\ \vdots\\ \Gamma_2, \exists z_{k-1}' \ldots z_1'.\, \widehat{\chi}''_{\mathrm{Sk}}[v/t] \vdash \Delta_2\end{array}}{\Gamma_2, \exists z_k' \ldots z_1'.\, \widehat{\chi}''_{\mathrm{Sk}}[v/t] \vdash \Delta_2}\,\exists{:}l\\ \Gamma_2, \forall v \exists z_k' \ldots z_1'.\, \widehat{\chi}''_{\mathrm{Sk}} \vdash \Delta_2\end{array}}{\Gamma_2, \exists y_1' \forall v \exists z_k' \ldots z_1'.\, \widehat{\chi}''_{\mathrm{Sk}} \vdash \Delta_2}\,\exists{:}l\\ \vdots\\ \Gamma_2, \exists y_{\ell-1}' \ldots y_1' \forall v \exists z_k' \ldots z_1'.\, \widehat{\chi}''_{\mathrm{Sk}} \vdash \Delta_2\end{array}}{\Gamma_2, \exists y_\ell' \ldots y_1' \forall v \exists z_k' \ldots z_1'.\, \widehat{\chi}''_{\mathrm{Sk}} \vdash \Delta_2}\,\exists{:}l$$

where π_2''' and $\widehat{\chi}''_{\mathrm{Sk}}$ and the terms $s_1, \ldots, s_k, t_1, \ldots, t_\ell$ are defined as follows. The terms
s_i $s_1, \ldots, s_k$ constitute an enumeration of all distinct *non-ground* terms that occur as arguments

in atoms $P(\ldots, s_i, \ldots)$ in $\widehat{\chi}'_{\mathrm{Sk}}$ and that contain at least one function symbol (a unary Skolem function from $\widehat{\varphi}_{\mathrm{Sk}}$). We assume that $s_1, \ldots, s_k$ are listed in descending order regarding their length, i.e. $\mathrm{len}(s_i) \geq \mathrm{len}(s_{i+1})$. Notice that, by Claim I, every s_i contains the variable v, only unary function symbols, and no constant symbols. Similarly, $t_1, \ldots, t_\ell$ constitutes an enumeration of all distinct *ground* terms (in descending order regarding term length) that occur as arguments in atoms in $\widehat{\chi}'_{\mathrm{Sk}}$. Furthermore, we assume the variables $y'_1, \ldots, y'_\ell, z'_1, \ldots, z'_k$ to be pairwise distinct, to be distinct from u, v, and to not occur in $\widehat{\varphi}_{\mathrm{Sk}}, \widehat{\chi}'_{\mathrm{Sk}}, \widehat{\psi}_{\mathrm{Sk}}, \Gamma_1, \Gamma_2, \Delta_1, \Delta_2$. The notation $\eta\big[s/t\big]$ is used to denote replacement of every occurrence of the term s by the term t in the formula η. We also use it to denote consistent replacement in derivations. Based on this notation, we define the formula $\widehat{\chi}''_{\mathrm{Sk}}$ to be t_i $\widehat{\chi}''_{\mathrm{Sk}}$

$$\widehat{\chi}''_{\mathrm{Sk}} := \widehat{\chi}'_{\mathrm{Sk}}\big[s_1/z'_1\big] \ldots \big[s_k/z'_k\big]\big[t_1/y'_1\big] \ldots \big[t_\ell/y'_\ell\big] \ .$$

Notice that all occurrences of v in $\widehat{\chi}'_{\mathrm{Sk}}$ that are an argument of a predicate symbol remain untouched. The derivation π_2''' results from π_2'' by replacing all occurrences of terms $s_i[v/t]$ π_2''' in π_2'' with z'_i and by replacing all occurrences of terms t_j with y'_j. The replacements are performed in the order $s_1[v/t], \ldots, s_k[v/t], t_1, \ldots, t_\ell$ (from left to right), i.e. we have

$$\pi_2''' := \pi_2''\big[s_1[v/t]/z'_1\big] \ldots \big[s_k[v/t]/z'_k\big]\big[t_1/y'_1\big] \ldots \big[t_\ell/y'_\ell\big] \ .$$

Notice that all occurrences of t in π_2''' that are an argument of a predicate symbol remain untouched.

As, by Claim III, the terms s_i, $s_i[v/t]$, and t_i do not share any function or constant symbols with terms occurring in Γ_2, Δ_2 that are important for the derivation of $\neg\widehat{\psi}_{\mathrm{Sk}}$, replacing these terms with (free) variables in π_2'' does not influence the parts of π_2'' that are important for deriving $\neg\widehat{\psi}_{\mathrm{Sk}}$ in the end. The new derivation steps in ρ'_2 using the rules introducing existential quantification on the left are sound, as Γ_2 and Δ_2 do not contain the variables y'_i and z'_i (which replace s_i and t_i, which in turn do not occur in Γ_2, Δ_2). Moreover, the new derivation steps in ρ'_1 using the rules introducing existential quantification on the right are sound, as the terms s_i only contain variables that do not occur bound in the antecedents.

We denote by ρ_1 the result of replacing occurrences of π'_1 in π_1 with appropriate occurrences ρ_1, ρ_2 of ρ'_1 and adapting the remaining proof parts accordingly (by replacing terms s_i by z'_i and t_i by y'_i wherever necessary). Similarly, the new version of π_2 is denoted by ρ_2. Let $\widehat{\chi}' := \widehat{\chi}''_{\mathrm{Sk}}$. Then, we can put the new derivations together to obtain $\widehat{\chi}'$

$$\dfrac{\widehat{\varphi}_{\mathrm{Sk}} \ \vdash \ \exists\bar{\mathrm{y}}'\forall v\exists\bar{\mathrm{z}}'.\,\widehat{\chi}' \quad (\rho_1) \qquad \exists\bar{\mathrm{y}}'\forall v\exists\bar{\mathrm{z}}'.\,\widehat{\chi}' \ \vdash \ \neg\widehat{\psi}_{\mathrm{Sk}} \quad (\rho_2)}{\widehat{\varphi}_{\mathrm{Sk}} \ \vdash \ \widehat{\psi}_{\mathrm{Sk}}}\ \mathrm{cut}(\exists\bar{\mathrm{y}}'\forall v\exists\bar{\mathrm{z}}'.\,\widehat{\chi}')$$

which constitutes a valid derivation representing the interpolation property we intended to prove. ◊

We observe that $\widehat{\varphi}_{\mathrm{Sk}} \models \widehat{\chi}$ is equivalent to $\widehat{\varphi}_{\mathrm{Sk}} \wedge \neg\widehat{\chi} \models \mathtt{false}$. De-Skolemization of the latter yields $\widehat{\varphi} \wedge \neg\widehat{\chi} \models \mathtt{false}$ or, equivalently, $\widehat{\varphi} \models \widehat{\chi}$. Also by de-Skolemization, from $\widehat{\chi} \wedge \widehat{\psi}_{\mathrm{Sk}} \models \mathtt{false}$ we conclude $\widehat{\chi} \wedge \neg\widehat{\psi} \models \mathtt{false}$.

Recall that $\widehat{\chi} = \exists\bar{\mathrm{y}}'\forall v\exists\bar{\mathrm{z}}'.\,\widehat{\chi}'$. We construct the formula χ' from $\widehat{\chi}'$ by simultaneously replacing χ' every literal $\neg P(\bar{s}\,)$ by $P(\bar{s}\,)$ and every literal $P(\bar{s}\,)$ by $\neg P(\bar{s}\,)$ for every $P \in \Pi_2$. Finally, we set $\chi := \exists\bar{\mathrm{y}}'\forall v\exists\bar{\mathrm{z}}'.\,\chi'$. χ

It remains to prove the following properties:

(4) $\varphi \models \chi$, and

(5) $\chi \wedge \neg\psi \models \mathtt{false}$.

Ad (4). Since $\big(\exists \bar{u} \forall v \exists \bar{w}.\, \widehat{\varphi'}\big) \wedge \neg\big(\exists \bar{y}' \forall v \exists \bar{z}'.\, \widehat{\chi'}\big) \models \texttt{false}$ holds if and only if $\big(\forall \bar{y} \exists x \forall \bar{z}.\, \varphi'\big) \wedge \neg\big(\exists \bar{y}' \forall v \exists \bar{z}'.\, \chi'\big) \models \texttt{false}$, and since the latter is equivalent to $\big(\forall \bar{y} \exists x \forall \bar{z}.\, \varphi'\big) \models \big(\exists \bar{y}' \forall v \exists \bar{z}'.\, \chi'\big)$, we in the end get $\varphi \models \chi$. ◊

Ad (5). As we observe that $\big(\exists \bar{y}' \forall v \exists \bar{z}'.\, \widehat{\chi'}\big) \wedge \neg\big(\forall \bar{y} \exists x \forall \bar{z}.\, \widehat{\psi'}\big) \models \texttt{false}$ holds if and only if $\big(\exists \bar{y}' \forall v \exists \bar{z}'.\, \chi'\big) \wedge \neg\big(\forall \bar{y} \exists x \forall \bar{z}.\, \psi'\big) \models \texttt{false}$, we in the end get

$$\big(\exists \bar{y}' \forall v \exists \bar{z}'.\, \widehat{\chi'}\big) \wedge \neg\big(\forall \bar{y} \exists x \forall \bar{z}.\, \widehat{\psi'}\big) \models \texttt{false} \quad \text{if and only if} \quad \chi \wedge \neg\psi \models \texttt{false}\ .$$

By (2), this yields $\chi \wedge \neg\psi \models \texttt{false}$. ◊

Because of the equivalence of $\chi \wedge \neg\psi \models \texttt{false}$ and $\chi \models \psi$, we have shown that χ is an AF sentence that satisfies Requirement (i) of the lemma.

Due to (3) and due to the way χ is constructed from N'_*, every positive occurrence of a predicate symbol P in χ entails the existence of a negative occurrence of P in $\neg\psi$, and every negative occurrence of a predicate symbol P in χ entails the existence of a positive occurrence of P in $\neg\psi$. Consequently, χ satisfies Requirement (ii) as well. □

Chapter 7

Beyond the Classical Decision Problem: Further Applications of Separateness

Evidently, the analysis of separateness of quantified variables and of weak dependences has applications in the quest for decidable first-order fragments. In the present chapter we briefly outline the applicability of these concepts to three other areas: the analysis of computational complexity of reasoning with respect to a fixed theory, e.g., rational arithmetic; proof complexity and automated reasoning in first-order logic; and the elimination of second-order quantifiers.

7.1 Separated Formulas and Linear Rational Arithmetic: A Little Case Study

In Chapter 4 we discussed fingerprints as a means to semantically characterize tuples of domain elements with respect to a given sentence and its atoms under a given structure. We recapitulate the underlying idea in Example 7.1.1. The purpose of the present section is to show that the number of fingerprints that possibly occur can be severely limited when we restrict the syntax of first-order formulas and focus on a certain kind of structures. As an exemplary case we study *linear rational arithmetic (LRA)*.

Example 7.1.1. *Consider the sentence* $\varphi := \forall x_1 \exists x_2 . P(x_1, x_2) \vee Q(x_1, x_2) \vee R(x_1, x_2)$ *and the structure* $\mathcal{A}$ *with* $\mathsf{A} := \{\mathsf{a}_1, \mathsf{a}_2, \mathsf{b}_P, \mathsf{b}_Q, \mathsf{b}_R, \mathsf{c}_{PQ}, \mathsf{c}_{QR}, \mathsf{c}_{PR}\}$ *and*

$$\begin{aligned} P^{\mathcal{A}} &:= \{\langle \mathsf{a}_1, \mathsf{b}_P \rangle, \langle \mathsf{a}_2, \mathsf{c}_{PQ} \rangle, \langle \mathsf{a}_2, \mathsf{c}_{PR} \rangle\} \ , \\ Q^{\mathcal{A}} &:= \{\langle \mathsf{a}_1, \mathsf{b}_Q \rangle, \langle \mathsf{a}_2, \mathsf{c}_{PQ} \rangle, \langle \mathsf{a}_2, \mathsf{c}_{QR} \rangle\} \ , \\ R^{\mathcal{A}} &:= \{\langle \mathsf{a}_1, \mathsf{b}_R \rangle, \langle \mathsf{a}_2, \mathsf{c}_{PR} \rangle, \langle \mathsf{a}_2, \mathsf{c}_{QR} \rangle\} \ . \end{aligned}$$

Let μ_2 *be the* fingerprint function *mapping pairs of elements from* $\mathcal{A}$*'s domain to fingerprints:*

$$\begin{aligned} \mu_2(\mathsf{a}_1, \mathsf{b}_P) &:= \{P(x_1, x_2)\} \ , \\ \mu_2(\mathsf{a}_1, \mathsf{b}_Q) &:= \{Q(x_1, x_2)\} \ , \\ \mu_2(\mathsf{a}_1, \mathsf{b}_R) &:= \{R(x_1, x_2)\} \ , \\ \mu_2(\mathsf{a}_2, \mathsf{c}_{PQ}) &:= \{P(x_1, x_2), Q(x_1, x_2)\} \ , \\ \mu_2(\mathsf{a}_2, \mathsf{c}_{PR}) &:= \{P(x_1, x_2), R(x_1, x_2)\} \ , \\ \mu_2(\mathsf{a}_2, \mathsf{c}_{RQ}) &:= \{R(x_1, x_2), Q(x_1, x_2)\} \ , \\ \mu_2(\mathsf{d}, \mathsf{e}) &:= \emptyset \quad \textit{for all other pairs } \langle \mathsf{d}, \mathsf{e} \rangle \in \mathsf{A}^2 \ . \end{aligned}$$

Expressed in technical terms, we have

$$\mu_2(\mathsf{d},\mathsf{e}) := \big\{A(x_1,x_2) \mid A(x_1,x_2) \text{ is an atom in } \varphi \text{ with } \mathcal{A} \models A(\mathsf{d},\mathsf{e})\big\} \ .$$

Fingerprints for unary tuples are a bit more interesting. Let μ_1 be the fingerprint function mapping elements from $\mathcal{A}$'s domain to fingerprints:

$$\begin{aligned}
\mu_1(\mathsf{a}_1) &:= \big\{\{P(x_1,x_2)\},\{Q(x_1,x_2)\},\{R(x_1,x_2)\}\big\} \ , \\
\mu_1(\mathsf{a}_2) &:= \big\{\{P(x_1,x_2),Q(x_1,x_2)\},\{P(x_1,x_2),R(x_1,x_2)\},\{R(x_1,x_2),Q(x_1,x_2)\}\big\} \ , \\
\mu_2(\mathsf{d}) &:= \{\emptyset\} \quad \text{for all other } \mathsf{d} \in \mathsf{A} \ .
\end{aligned}$$

Technically, we have $\mu_1(\mathsf{d}) := \big\{S \bigm| S = \mu_2(\mathsf{d},\mathsf{d}')$ for some element $\mathsf{d}' \in \mathsf{A}\big\}$. Hence, $\mu_1(\mathsf{d})$ characterizes for cases where the value d is assigned to x_1 which atoms will potentially become true under $\mathcal{A}$ when choosing a values for x_2 from $\mathcal{A}$'s domain.

It is easy to see that if we were to consider a different structure $\mathcal{B}$ the image of μ_2 could potentially contain a number of fingerprints that is exponential in the number of atoms occurring in φ, i.e. up to $2^3 = 8$ sets of atoms. For the image of μ_1, we even get a doubly exponential upper bound, i.e. $2^{2^3} = 256$. Indeed, we have (almost) constructed such structures in the proofs of Theorems 3.2.7, 3.9.9, 3.10.8, and 3.12.5.

The fingerprints described in the above example are very general. This generality is certainly necessary in settings where we do not restrict the class of structures we consider. However, there are (classes of) structures that do not require as much freedom for the fingerprints of domain elements. In such settings fingerprints could be structured in a simpler way. The example we shall consider is *linear arithmetic over the rationals*. More precisely, we consider first-order formulas over the vocabulary $\Sigma_{\mathrm{LRA}} := \langle\{<,\leq,\neq,\geq,>\},\mathbb{Q}\cup\{+,\cdot\}\rangle$ containing the rational numbers as constant symbols under the following syntactic restriction: multiplication is only allowed in terms $a \cdot x$ where a is some rational coefficient and x is some first-order variable. We shall call the set of all first-order formulas in which all terms are LRA terms *LRA formulas*. Semantically, we fix the structure under which LRA formulas are interpreted to $\mathbb{Q}$, the rational numbers with the standard interpretation for the symbols in Σ_{LRA}. For convenience, we use abbreviations such as $\frac{3}{2}x - 5y - z$ for the formal expression $\frac{3}{2}\cdot x + (-5)\cdot y + (-1)\cdot z$.

LRA formulas

Example 7.1.2. *Consider the linear-arithmetic atoms $A(x,y) := x - y \leq 1$, $B(x,y) := \frac{1}{2}x + y \geq 2$, $C(x,y) := \frac{1}{3}x - y \geq \frac{1}{3}$, cf. Figure 7.1. In analogy to Example 7.1.1 we define two fingerprint functions μ_1,μ_2 with respect to the structure $\mathbb{Q}$ as follows:*

$$\mu_2(\mathsf{d},\mathsf{e}) := \big\{D(x,y) \mid D(x,y) \text{ is one of the above atoms and } \mathbb{Q} \models D(\mathsf{d},\mathsf{e})\big\}$$

and

$$\mu_1(\mathsf{d}) := \big\{S \bigm| S = \mu_2(\mathsf{d},\mathsf{e}) \text{ for some element } \mathsf{e} \in \mathbb{Q}\big\} \ .$$

The image of μ_2 contains seven elements, namely, $\emptyset,\{A\},\{B\},\{C\},\{A,B\},\{A,C\},\{B,C\}$. The image of μ_1, on the other hand, contains only four elements, namely, $\big\{\{A,B\},\{A\},\{A,C\},\{C\}\big\}$, $\big\{\{A,B\},\{A\},\emptyset,\{C\}\big\}$, $\big\{\{A,B\},\{B\},\emptyset,\{C\}\big\}$, $\big\{\{A,B\},\{B\},\{B,C\},\{C\}\big\}$.

The above example illustrates nicely that fingerprints over linear-arithmetic atoms are very restricted. For instance, there cannot be any rational number r with a fingerprint $\mu_1(r)$ that is a superset of $\big\{\{A(x,y)\},\{A(x,y),B(x,y)\},\{B(x,y)\},\{C(x,y)\}\big\}$. The reason is that for any fixed r the solution spaces for $A(r,y)$, $B(r,y)$, and $C(r,y)$ are either empty or singleton sets or the union of (at most two) unbounded intervals, respectively. Hence, if there are three rationals s_1,s_2,s_3 such that we have

$$\begin{aligned}
\mathbb{Q} &\models \ \ A(r,s_1) \wedge \neg B(r,s_1) \wedge \neg C(r,s_1) \ , \\
\mathbb{Q} &\models \ \ A(r,s_2) \wedge \ \ B(r,s_2) \wedge \neg C(r,s_2) \ , \\
\mathbb{Q} &\models \neg A(r,s_3) \wedge \ \ B(r,s_3) \wedge \neg C(r,s_3) \ ,
\end{aligned}$$

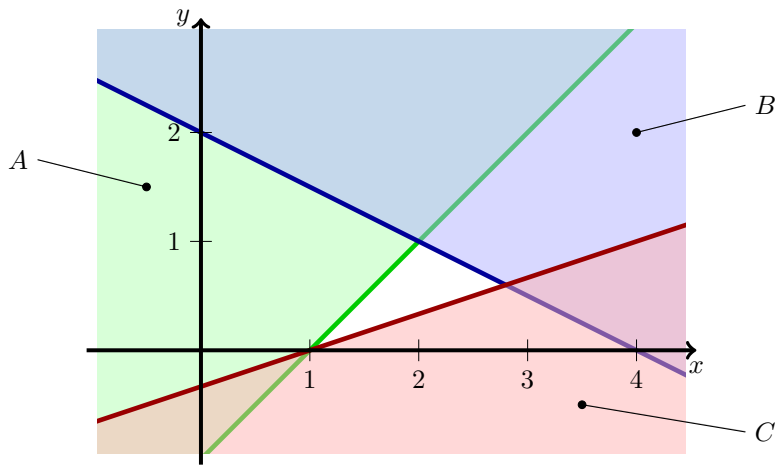
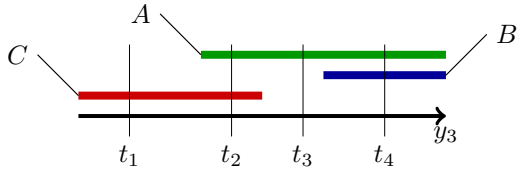


Figure 7.1: Illustration of the solution sets of the three linear-arithmetic atoms $A(x, y) = x - y \leq 1$ (green), $B(x, y) = \frac{1}{2}x + y \geq 2$ (blue), $C(x, y) := \frac{1}{3}x - y \geq \frac{1}{3}$ (red).

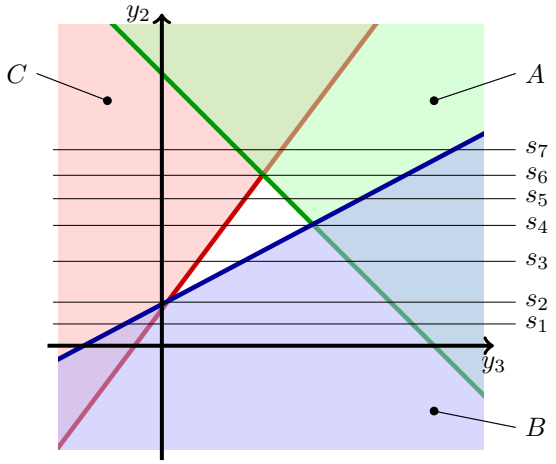
then the solution spaces for $A(r, y)$ and $B(r, y)$ with respect to the variable y are (supersets of) unbounded intervals, respectively, such that their union covers the complete rational axis. This entails that there cannot be any s' such that $\mathbb{Q} \models \neg A(r, s') \wedge \neg B(r, s') \wedge C(r, s')$.

This peculiarity of linear-arithmetic atoms severely restricts the fingerprints that can possibly occur. We shall derive an upper bound on the number of distinct fingerprints that is doubly exponential in the number of occurring variables in Lemma 7.1.5. In order to do so, we will leverage methods and results from the field of *quantifier elimination*. But before we develop the formal argument, Figures 7.2–7.5 informally describe some more intuitions about fingerprints for linear rational arithmetic while suggesting a more suitable notation for linear-arithmetic fingerprints.



$\nu_3(r, s, t_1) = \{C\}$
$\nu_3(r, s, t_2) = \{A, C\}$
$\nu_3(r, s, t_3) = \{A\}$
$\nu_3(r, s, t_4) = \{A, B\}$

Figure 7.2: Illustration of the solution set of three linear-arithmetic atoms $A(y_1, y_2, y_3)$ (in green), $B(y_1, y_2, y_3)$ (in blue), and $C(y_1, y_2, y_3)$ (in red) for certain fixed values r for y_1 and s for y_2. The thin vertical lines mark values yielding the four possible fingerprints $\nu_3(r, s, t)$ for any rational $t \in \mathbb{Q}$ that is assigned to y_3. The overall fingerprint $\nu_2(r, s)$ is represented by the sequence $\{C\}|\{A\}\{C\}\{B\}$. Intuitively, we construct $\nu_2(r, s) = S_0|S_1S_2S_3$ as follows. Traversing the y_3-axis from $-\infty$ to $+\infty$ for fixed $y_1 = r$, $y_2 = s$, we start in a situation where C but neither A nor B are satisfied under $\mathbb{Q}$. Hence, we set $S_0 := \{C\}$. On our traversal of the axis, we first encounter some point at which A changes its truth value — it becomes `true` (and will stay so for the rest of our journey along the y_3-axis). Since exclusively the truth value of A changes at this point, we append the set $S_1 := \{A\}$. Going on in this direction, we next encounter a point where the truth value of C changes from `true` to `false`. Since A and B keep their respective truth values, we append the set $S_2 := \{C\}$. Like for A, the truth value of C will remain unchanged for the rest of our traversal of the axis. Finally, we reach some point at which the truth value of B flips from `false` to `true`. Therefore, we append $S_3 := \{B\}$. As nothing will change when traversing the rest of the y_3-axis, the construction of $\nu_2(r, s)$ is complete.



$\nu_2(r, s_7) = \{C\}|\{A\}\{C\}\{B\}$
$\nu_2(r, s_6) = \{C\}|\{A, C\}\{B\}$
$\nu_2(r, s_5) = \{C\}|\{C\}\{A\}\{B\}$
$\nu_2(r, s_4) = \{C\}|\{C\}\{A, B\}$
$\nu_2(r, s_3) = \{C\}|\{C\}\{B\}\{A\}$
$\nu_2(r, s_2) = \{C\}|\{B, C\}\{A\}$
$\nu_2(r, s_1) = \{C\}|\{B\}\{C\}\{A\}$

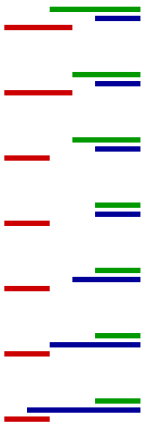


Figure 7.3: Illustration of the solution set of three linear-arithmetic atoms $A(y_1, y_2, y_3)$ (in green), $B(y_1, y_2, y_3)$ (in blue), and $C(y_1, y_2, y_3)$ (in red) for some fixed value r for y_1. The thin horizontal lines mark values yielding the seven possible fingerprints $\nu_2(r, s)$ for any rational $s \in \mathbb{Q}$ that is assigned to y_2. The overall fingerprint $\nu_1(r)$ is represented by the sequence $\big(\{C\}|\{B\}\{C\}\{A\}\big)\big|\{B, C\}\{A, B\}\{A, C\}$, where the initial subsequence $\{C\}|\{B\}\{C\}\{A\}$ describes the fingerprint at $y_3 = s_1$ and the sets $\{B, C\}$, $\{A, B\}$, $\{A, C\}$ originate from the three intersection points of the hyperplanes represented by A, B, C, respectively, when the relation symbols in A, B, C are replaced with equality. When traversing the y_2-axis from $-\infty$ to $+\infty$, we meet the intersection points $\{B, C\}$ at $y_2 = s_2$, $\{A, B\}$ at $y_2 = s_4$, and $\{A, C\}$ at $y_2 = s_6$.

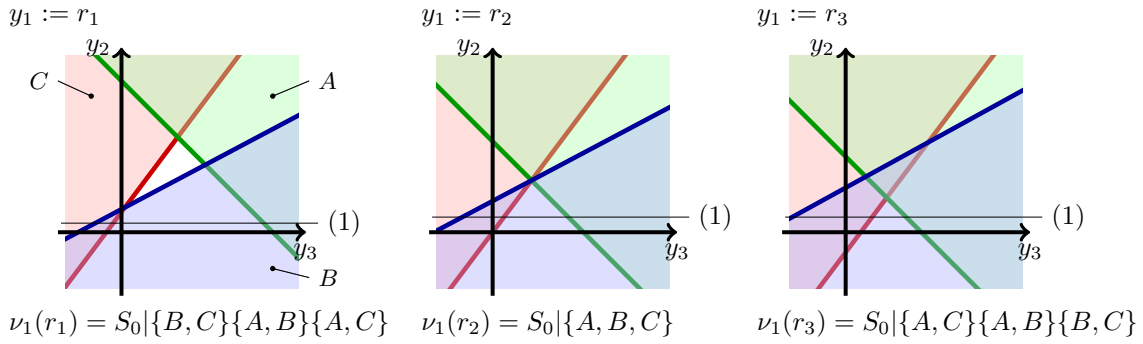


Figure 7.4: Illustration of the solution set of three linear-arithmetic atoms $A(y_1, y_2, y_3)$ (in green), $B(y_1, y_2, y_3)$ (in blue), and $C(y_1, y_2, y_3)$ (in red) for certain rational values $r_1 < r_2 < r_3$ assigned to y_1. Each diagram represents one of the fingerprints $\nu_1(r_1)$, $\nu_1(r_2)$, $\nu_1(r_3)$, as indicated. The initial sequence S_0 for each of these fingerprints is the sequence $\{C\}|\{B\}\{C\}\{A\}$, i.e. the fingerprint associated with (1). After S_0 each fingerprint lists the intersections of the hyperplanes induced by A, B, C in the order they occur, viewed from $-\infty$ along the y_2-axis in the positive direction. Every set in the list represents a point in which the indicated hyperplanes intersect. The overall fingerprint ν_0 is represented by the sequence $\Big(\big(\{C\}|\{B\}\{C\}\{A\}\big)\big|\{B,C\}\{A,B\}\{A,C\}\Big)\Big|\{A,B,C\}$.

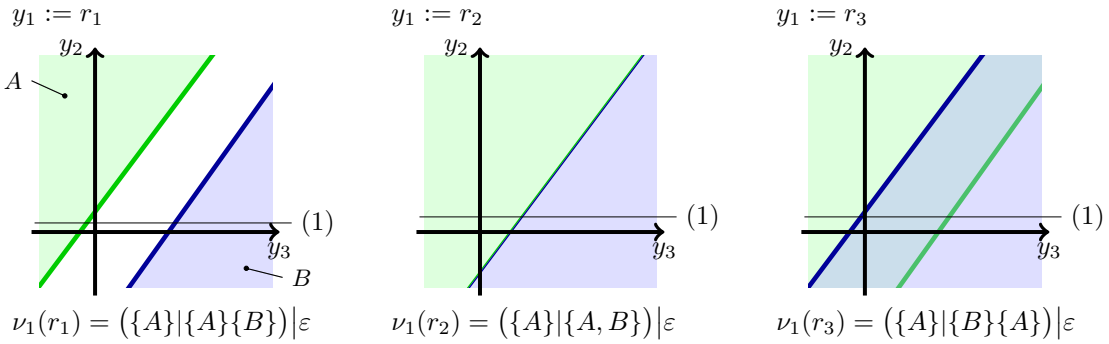


Figure 7.5: Illustration of the solution set of two linear-arithmetic atoms $A(y_1, y_2, y_3)$ (in green) and $B(y_1, y_2, y_3)$ (in blue) for certain rational values $r_1 < r_2 < r_3$ assigned to y_1. Each diagram represents one of the fingerprints $\nu_1(r_1)$, $\nu_1(r_2)$, $\nu_1(r_3)$, as indicated. The initial sequence in the fingerprint representations differ for each of the three points, however, after the initial sequence, the empty sequence ε follows. This means, for fixed $y_1 = r$ the fingerprints $\nu_2(r, y_2)$ along the y_2-axis do not change. The overall fingerprint ν_0 is represented by the sequence $\Big(\big(\{A\}|\{A\}\{B\}\big)|\varepsilon\Big)\Big|\{A,B\}$.

Next, we describe the *virtual substitution* method for the elimination of existential quantifiers in first-order formulas over linear rational arithmetic. We have already outlined the method in Chapter 2 (page 20). In our simple setting virtual substitution is based on two components: (a) a method for extracting a set of *testpoints* from a given formula φ, also called an *elimination set*, and (b) the virtual substitution operator $\big[x/\!\!/t\big]$, which is a generalization of the syntactic substitution operator $\big[x/t\big]$. An LRA formula is called *positive*, if it does not contain any negation sign. Notice that any LRA formula can be transformed into an equivalent positive LRA formula whose length

virtual substitution

positive formulas

is linear in the length of the original. For example, every negative literal $\neg\, s \leq t$ can be replaced by the positive $s > t$; the relation symbol "$\neq$" is treated as first-class citizen, i.e. $s \neq t$ is *not* an abbreviation for $\neg\, s = t$. Let At be a finite set of LRA atoms of the form $a_1 z_1 + \ldots a_n z_n \lhd b$ where the a_i and b are rational coefficients, the z_i are first-order variables, and $\lhd \in \{<, \leq, =, \neq, \geq, >\}$. For every z_i the *elimination set* $\mathrm{Elim}_{z_i}(\mathrm{At})$ is the smallest set of *formal terms* satisfying the following properties:

- At
- $\mathrm{Elim}_z(\mathrm{At})$

(i) $-\infty \in \mathrm{Elim}_{z_i}(\mathrm{At})$.

(ii) For every atom $(a_1 z_1 + \ldots + a_n z_n \lhd b) \in \mathrm{At}$ with $a_i \neq 0$ we set
$$t := \tfrac{1}{a_i}\big(b - a_1 z_1 + \ldots + a_{i-1} z_{i-1} + a_{i+1} z_{i+1} + \ldots + a_n z_n\big).$$
We require that

if $a_i > 0$ and $\lhd \in \{=, \geq\}$, then $t \in \mathrm{Elim}_{z_i}(\mathrm{At})$,
if $a_i < 0$ and $\lhd \in \{=, \leq\}$, then $t \in \mathrm{Elim}_{z_i}(\mathrm{At})$,
if $a_i > 0$ and $\lhd \in \{\neq, >\}$, then $(t + \varepsilon) \in \mathrm{Elim}_{z_i}(\mathrm{At})$,
if $a_i < 0$ and $\lhd \in \{\neq, <\}$, then $(t + \varepsilon) \in \mathrm{Elim}_{z_i}(\mathrm{At})$.

The symbol ε is used as a "dummy symbol for [...] a positive infinitesimal" [LW93], page 454. Notice that for any z_i the number of testpoints in $\mathrm{Elim}_{z_i}(\mathrm{At})$ is at most linear in the number of atoms that belong to At. Moreover, only atoms contribute to $\mathrm{Elim}_{z_i}(\mathrm{At})$ in which z_i occurs with a non-zero coefficient and that entail some kind of lower bound for z_i. We could, dually, restrict our attention to upper bounds instead of lower bounds [LW93, HVW17a].

- $A\big[z /\!\!/ t\big]$

Given any testpoint t, the application of the virtual substitution operator $\big[z_i /\!\!/ t\big]$ to LRA atoms is defined as follows. Consider any atom $A(z_1, \ldots, z_n)$ of the form $a_1 z_1 + \ldots + a_n z_n \lhd b$ with $a_i = 0$. Then, we set $A[z_i /\!\!/ t] := A$. Consider any atom $A(z_1, \ldots, z_n)$ that is equivalent to $z_i \lhd s$ for some LRA expression s of the form $s := \frac{1}{a_i}\big(b - a_1 z_1 + \ldots + a_{i-1} z_{i-1} + a_{i+1} z_{i+1} + \ldots + a_n z_n\big)$ with $a_i \neq 0$ and where $\lhd \in \{<, \leq, =, \neq, \geq, >\}$. We then define

$$A\big[z_i /\!\!/ t\big] := t \lhd s \qquad \text{if } t \neq -\infty \text{ and } t \neq t' + \varepsilon \text{ for any } t',$$

$$A\big[z_i /\!\!/ -\infty\big] := \begin{cases} \texttt{true} & \text{if } \lhd \in \{<, \leq, \neq\}, \\ \texttt{false} & \text{if } \lhd \in \{=, \geq, >\}, \end{cases}$$

$$A\big[z_i /\!\!/ t' + \varepsilon\big] := \begin{cases} t' < s & \text{if } \lhd \in \{<, \leq\}, \\ t' \geq s & \text{if } \lhd \in \{\geq, >\}, \\ \texttt{false} & \text{if } \lhd \in \{=\}, \\ \texttt{true} & \text{if } \lhd \in \{\neq\}. \end{cases}$$

For technical reasons, we assume that necessary normalization steps are done implicitly, i.e. in order to apply the operator $\big[z /\!\!/ t\big]$ to any LRA atom A, the atom need not be in the form $a_1 z_1 + \ldots + a_n z_n \lhd b$. It is only required that A has an equivalent in this syntactic form. This is certainly true for the atoms `true` and `false`, for example, as the former is equivalent to $0 z_1 + \ldots + 0 z_n \leq 0$ and the latter is equivalent to $0 z_1 + \ldots + 0 z_n < 0$. For compound LRA formulas the virtual substitution operator shall behave like the usual operator for syntactic substitution.

Loos and Weispfenning have shown that virtual substitution can be used to eliminate existential quantifiers from LRA formulas.

Proposition 7.1.3 ([LW93], Theorem 3.5)**.** *Consider any positive LRA formula $\exists z_i.\, \psi(\bar{z})$ with quantifier-free $\psi(\bar{z})$. Let* At *be the set of all atoms that occur in $\psi(\bar{z})$ normalized so that each has the form $a_1 z_1 + \ldots a_n z_n \lhd b$. Then, we have*

$$\mathbb{Q} \models \big(\exists z_i.\, \psi(\bar{z})\big) \longleftrightarrow \bigvee_{t \,\in\, \mathrm{Elim}_{z_i}(\mathrm{At})} \psi\big[z_i /\!\!/ t\big] \ .$$

The quantifier elimination technique in Proposition 7.1.3 makes use of the fact that existential quantification can be conceived as a disjunction over all domain elements. Innermost existential quantifiers are replaced by finite (syntactic) disjunction over a finite set of elimination terms. This is done in a way that implicitly constructs a Skolem function using only specific terms that can be derived from the formula at hand. This implicit Skolem function includes an unspecific case split: one of the "proposed" solutions will work. An explicitly formulated function, however, would be more specific about which solution has to be used in what case.

In what follows, we use Proposition 7.1.3 and our observations regarding elimination sets to formally derive an upper bound regarding the number of fingerprints that appear with respect to a given LRA sentence. The missing link between the two concepts of fingerprints and quantifier elimination by means of virtual substitution is a representation of fingerprints by first-order sentences. Let At be a finite set of LRA atoms of the form $a_1 z_1 + \ldots + a_n z_n \lhd b$ where the a_i and b are rational coefficients and $\lhd \in \{<, \leq, =, \neq, \geq, >\}$. Let Ψ_n be the set of all formulas of the form $\exists z_n.\ \bigwedge_{A \in \mathrm{At}} [\neg] A(z_1, \ldots, z_n)$, where $[\neg]\psi$ stands as placeholder for a negated or non-negated formula. For every i with $1 \leq i < n$ let Ψ_i be the set of all formulas of the form $\exists z_i.\ \bigwedge_{\psi \in \Psi_{i+1}} [\neg]\psi(z_1, \ldots, z_i)$. Then, every set Ψ_i contains $2^{\uparrow n-i+1}(|\mathrm{At}|)$ formulas. In each of these formulas the variables $z_1, \ldots, z_{i-1}$ occur freely. Ψ_i

Remark 7.1.4. *It is easy to see that any formula $\psi(z_1, \ldots, z_{i-1})$ in any Ψ_i represents exactly one possible fingerprint for tuples $\langle r_1, \ldots, r_{i-1}\rangle$ of rationals. The formula sets Ψ_i emphasize the close relationship between the notion of fingerprint and the standard model-theoretic notion of* type.[1] *In model theory types are usually used to investigate expressiveness of full first-order logic with respect to certain structures. Hence, the atoms from which types are built are often only restricted by the considered vocabulary, the variables that may occur freely, and limits imposed on the quantifier rank. The sets Ψ_i, on the other hand, are limited to certain forms of atoms, namely the ones given in* At. *Hence, it may happen that an atom $P(x_1, x_2, x_3)$ plays a role in Ψ_i and $P(x_1, x_1, x_1)$ does not. In the present text such fine-grained distinctions make sense, since they allow us to focus on the expressiveness of concrete sentences — e.g. with respect to dependences between quantified variables — rather than expressiveness of the whole language of first-order logic over a given vocabulary. Such fine-grained considerations do not seem to be common practice in (finite) model theory.*

Let $\widetilde{\mathrm{At}}$ be the set $\widetilde{\mathrm{At}} := \mathrm{At} \cup \{\overline{A} \mid A \in \mathrm{At}\}$, where the atom $\overline{A}$ is the *converse of* A, e.g. $\overline{(s \leq t)} = (s > t)$, $\overline{(s = t)} = (s \neq t)$, $\overline{(s < t)} = (s \geq t)$. Let $\widetilde{\mathrm{At}}_{-\langle z_j, \ldots, z_n\rangle}$ be the result of eliminating $z_n, z_{n-1}, \ldots, z_j$ one after the other from the atoms in $\widetilde{\mathrm{At}}$ by means of virtual substitution. Formally, $\widetilde{\mathrm{At}}_{-\langle z_j, \ldots, z_n\rangle}$ is defined as follows: $\widetilde{\mathrm{At}}$, $\overline{A}$ $\widetilde{\mathrm{At}}_{-\langle z_j, \ldots, z_n\rangle}$

$$\widetilde{\mathrm{At}}_{-\langle z_n\rangle} := \left\{ A\big[z_n /\!\!/ t\big], \overline{A\big[z_n /\!\!/ t\big]} \;\middle|\; A \in \widetilde{\mathrm{At}} \text{ and } t \in \mathrm{Elim}_{z_n}\big(\widetilde{\mathrm{At}}\big) \right\} \text{, and}$$
$$\widetilde{\mathrm{At}}_{-\langle z_i, \ldots, z_n\rangle} := \left\{ A\big[z_i /\!\!/ t\big], \overline{A\big[z_i /\!\!/ t\big]} \;\middle|\; A \in \widetilde{\mathrm{At}}_{-\langle z_{i+1}, \ldots, z_n\rangle} \text{ and } t \in \mathrm{Elim}_{z_i}\big(\widetilde{\mathrm{At}}_{-\langle z_{i+1}, \ldots, z_n\rangle}\big) \right\} .$$

Since we know that the number of testpoints in any elimination set $\mathrm{Elim}_{z_i}(S)$ is at most linear in the cardinality of S, there must be some positive constant c such that we observe

$$\begin{aligned}
\big|\widetilde{\mathrm{At}}_{-\langle z_n\rangle}\big| &\leq 2c \cdot |\widetilde{\mathrm{At}}|^2 \text{ ,} \\
\big|\widetilde{\mathrm{At}}_{-\langle z_{n-1}, z_n\rangle}\big| &\leq 2c \cdot \big(2c \cdot |\widetilde{\mathrm{At}}|^2\big)^2 \text{ ,} \\
&\vdots \\
\big|\widetilde{\mathrm{At}}_{-\langle z_i, \ldots, z_n\rangle}\big| &\leq (2c)^{2^{n-i+1}} \cdot |\widetilde{\mathrm{At}}|^{2^{n-i+1}} \leq 2^{(\log 2c) \cdot 2^{n-i+1}} \cdot 2^{(\log |\mathrm{At}|) \cdot 2^{n-i+1}} = 2^{(\log 2c + \log |\mathrm{At}|) \cdot 2^{n-i+1}} .
\end{aligned}$$

For every i, $1 \leq i \leq n$, and any sequence $\bar{r} := r_1, \ldots, r_{i-1}$ of rationals let the set $\widehat{\Psi}_i(\bar{r})$ be the set of formulas $\psi(z_1, \ldots, z_{i-1}) \in \Psi_i$ for which we have $\mathbb{Q} \models \psi(\bar{r})$. In the beginning of the $\widehat{\Psi}_i(\bar{r})$

[1] Compare, e.g., the notion of *type* in [CK90] or *rank-k type* in [Lib04]. Another close relative is the notion of constituent by Hintikka, see [Hin65, Ran87]

present section we explained intuitively why the set of actually occurring fingerprints with respect to LRA formulas is very limited compared to the general case. The next lemma will show a similar discrepancy between the sets Ψ_i and $\widehat{\Psi}_i(\bar{r})$. More precisely, only comparatively few formulas from Ψ_i are satisfied under $\mathbb{Q}$ for certain tuples $\bar{r}$ of rationals. We shall then link these two observations in Lemma 7.1.8.

Lemma 7.1.5. *For any fixed sequence $\bar{r} := r_1, \ldots, r_{i-1}$ of rationals we have $\big|\widehat{\Psi}_i(\bar{r})\big| \in \mathcal{O}\big(2^{(d+\log|\mathrm{At}|)\cdot 2^{n-i}}\big)$ for some positive constant d.*

Proof. Let $\beta := [z_1 \mapsto r_1, \ldots, z_{i-1} \mapsto r_{i-1}]$.

Claim I: For any two distinct formulas $\exists z_i.\, \psi_1(z_1, \ldots, z_i)$ and $\exists z_i.\, \psi_2(z_1, \ldots, z_i)$ from $\widehat{\Psi}_i(\bar{r})$ and all distinct rationals q_i, q_i' we observe that $\mathbb{Q}, \beta[z_i \mapsto q_i] \models \psi_1(z_1, \ldots, z_i)$ together with $\mathbb{Q}, \beta[z_i \mapsto q_i'] \models \psi_2(z_1, \ldots, z_i)$ entails $q_i \neq q_i'$.

Proof: By definition of Ψ_i, ψ_1 and ψ_2 are conjunctions of formulas $(\exists z_i.\ \ldots)$ such that, without loss of generality, ψ_1 contains at least one conjunct $\neg\varphi$ with $\varphi \in \Psi_{i+1}$ while ψ_2 contains the conjunct φ. But then, we have $\mathbb{Q}, \beta[z_i \mapsto q_i] \models \neg\varphi(z_1, \ldots, z_i)$ and $\mathbb{Q}, \beta[z_i \mapsto q_i'] \models \varphi(z_1, \ldots, z_i)$, which entails $q_i \neq q_i'$. ◊

ψ_1', ψ_2'

Consider any two distinct formulas $\exists z_i.\, \psi_1(z_1, \ldots, z_i)$ and $\exists z_i.\, \psi_2(z_1, \ldots, z_i)$ from $\widehat{\Psi}_i(\bar{r})$. Let ψ_1' be the result of first transforming ψ_1 into negation normal form and then replacing every atom $\neg A$ by its converse $\overline{A}$. Let ψ_2' be defined analogously, starting from ψ_2. Then, ψ_1' and ψ_2' are positive formulas and $\mathbb{Q}$-equivalent to ψ_1 and ψ_2, respectively. Moreover, all atoms occurring in ψ_1' and ψ_2' belong to $\widetilde{\mathrm{At}}$. Using virtual substitution, all existential quantifiers in the formulas $\psi_1'(z_1, \ldots, z_i)$ and $\psi_2'(z_1, \ldots, z_i)$ can be eliminated, yielding quantifier-free formulas

ψ_1'', ψ_2''

$\psi_1''(z_1, \ldots, z_i)$ and $\psi_2''(z_1, \ldots, z_i)$ over the atoms in $\widetilde{\mathrm{At}}_{-\langle z_{i+1}, \ldots, z_n\rangle}$ that are $\mathbb{Q}$-equivalent to ψ_1 and ψ_2, respectively. With another application of virtual substitution, the existential quantifier $\exists z_i$ in both $\exists z_i.\, \psi_1''(z_1, \ldots, z_i)$ and $\exists z_i.\, \psi_2''(z_1, \ldots, z_i)$ can be eliminated, using only a subset of the testpoints from $\mathrm{Elim}_{z_i}\big(\widetilde{\mathrm{At}}_{-\langle z_{i+1}, \ldots, z_n\rangle}\big)$, which is a set of size

$$\big|\mathrm{Elim}_{z_i}\big(\widetilde{\mathrm{At}}_{-\langle z_{i+1}, \ldots, z_n\rangle}\big)\big| \in \mathcal{O}\big(2^{(d+\log|\mathrm{At}|)\cdot 2^{n-i}}\big)$$

for some positive constant d. Put more precisely, $\exists z_i.\, \psi_1''(z_1, \ldots, z_i)$ is $\mathbb{Q}$-equivalent to

$$\bigvee_{t \,\in\, \mathrm{Elim}_{z_i}(\widetilde{\mathrm{At}}_{-\langle z_{i+1}, \ldots, z_n\rangle})} \psi_1''\big[z_i /\!\!/ t\big]$$

and $\exists z_n.\, \psi_2''(z_1, \ldots, z_n)$ is $\mathbb{Q}$-equivalent to

$$\bigvee_{t \,\in\, \mathrm{Elim}_{z_i}(\widetilde{\mathrm{At}}_{-\langle z_{i+1}, \ldots, z_n\rangle})} \psi_2''\big[z_i /\!\!/ t\big]\ .$$

By virtue of Claim I, for every testpoint $t \in \mathrm{Elim}_{z_i}\big(\widetilde{\mathrm{At}}_{-\langle z_{i+1}, \ldots, z_n\rangle}\big)$ we have that $\mathbb{Q}, \beta \models \psi_1''\big[z_i /\!\!/ t\big]$ entails $\mathbb{Q}, \beta \not\models \psi_2''\big[z_i /\!\!/ t\big]$. Consequently, for every $\exists z_i.\, \psi$ in $\widehat{\Psi}_i(\bar{r})$ and ψ's quantifier-free equivalent ψ' (obtained by means of virtual substitution) there is at least one testpoint $t \in \mathrm{Elim}_{z_i}\big(\widetilde{\mathrm{At}}_{-\langle z_{i+1}, \ldots, z_n\rangle}\big)$ such that $\mathbb{Q}, \beta \models \psi'\big[z_i /\!\!/ t\big]$ and $\mathbb{Q}, \beta \not\models \varphi'\big[z_i /\!\!/ t\big]$ for every $\exists z_i.\, \varphi$ in $\widehat{\Psi}_i(\bar{r})$ with $\varphi \neq \psi$ and φ's quantifier-free equivalent φ'. This entails $\big|\widehat{\Psi}_i(\bar{r})\big| \leq \big|\mathrm{Elim}_{z_i}\big(\widetilde{\mathrm{At}}_{-\langle z_{i+1}, \ldots, z_n\rangle}\big)\big| \in \mathcal{O}\big(2^{(d+\log|\mathrm{At}|)\cdot 2^{n-i}}\big)$ for some positive constant d. □

Corollary 7.1.6. *The number of sentences in Ψ_1 that are satisfied under $\mathbb{Q}$ is at most doubly exponential in n and at most polynomial in the number of atoms in* At.

Now consider any LRA sentence $\varphi := \forall x_1 \exists y_1 \ldots \forall x_n \exists y_n.\, \psi(\bar{\mathrm{x}}, \bar{\mathrm{y}})$ with quantifier-free ψ in which the tuples $\bar{\mathrm{x}} := \langle x_1, \ldots, x_n \rangle$ and $\bar{\mathrm{y}} := \langle y_1, \ldots, y_n \rangle$ are separated. Without loss of generality, we assume that φ is a positive formula and that every atom in it has the form $a_1 z_1 + \ldots a_n z_n \lhd b$. Let At be the set of atoms occurring in φ. Then, At can be partitioned into two parts $\mathrm{At}_{\bar{\mathrm{x}}}$ and $\mathrm{At}_{\bar{\mathrm{y}}}$ such that $\mathrm{vars}(\mathrm{At}_{\bar{\mathrm{x}}}) \subseteq \bar{\mathrm{x}}$ and $\mathrm{vars}(\mathrm{At}_{\bar{\mathrm{y}}}) \subseteq \bar{\mathrm{y}}$.

Definition 7.1.7 (Fingerprint functions μ_ℓ)**.** *We define the family of* fingerprint functions μ_ℓ *with $0 \leq \ell \leq n$ in analogy to Definition 4.2.4 as follows:*

$\mu_n : \mathbb{Q}^n \to \mathcal{P}\mathrm{At}_{\bar{\mathrm{x}}}$ *such that for every sequence $\bar{r} \in \mathbb{Q}^n$ and every $A(\bar{\mathrm{x}}) \in \mathrm{At}_{\bar{\mathrm{x}}}$ we have $A(\bar{\mathrm{x}}) \in \mu_n(\bar{r})$ if and only if $\mathbb{Q} \models A(\bar{r})$.*

For every ℓ, $1 \leq \ell < n$, we set

$\mu_\ell : \mathbb{Q}^\ell \to \mathcal{P}^{n-\ell+1}\mathrm{At}_{\bar{\mathrm{x}}}$ *such that for every sequence $\bar{r} \in \mathbb{Q}^\ell$ and every $S \in \mathcal{P}^{n-\ell}\mathrm{At}_{\bar{\mathrm{x}}}$ we have $S \in \mu_\ell(\bar{r})$ if and only if there is some rational number q such that $\mu_{\ell+1}(\bar{r}, q) = S$.*

Moreover, we define

$\mu_0 \subseteq \mathcal{P}^{n+1}\mathrm{At}_{\bar{\mathrm{x}}}$ *such that for every $S \in \mathcal{P}^n\mathrm{At}_{\bar{\mathrm{x}}}$ we have $S \in \mu_0$ if and only if there is some rational number q such that $\mu_1(q) = S$.*

We denote the image *of a fingerprint function μ_ℓ by $\mathrm{im}(\mu_\ell) := \{\mu_\ell(\bar{r}) \mid \bar{r} \in \mathbb{Q}^\ell\}$.*

Lemma 7.1.8. *For every i, $0 \leq i \leq n-1$, and any fixed sequence $\bar{r} \in \mathbb{Q}^i$ we have $|\mu_i(\bar{r})| \in \mathcal{O}(2^{(d+\log|\mathrm{At}_{\bar{\mathrm{x}}}|)\cdot 2^{n-i}})$ for some positive constant d.*

Proof. We define the sets $\Psi_i^{\bar{\mathrm{x}}}$ and $\widehat{\Psi}_i^{\bar{\mathrm{x}}}(\bar{r})$ in analogy to the sets Ψ_i and $\widehat{\Psi}_i(\bar{r})$ based on $\mathrm{At}_{\bar{\mathrm{x}}}$ rather than an arbitrary set At. Moreover, let the bijections $\rho_i : \mathcal{P}^{n-i+1}\mathrm{At}_{\bar{\mathrm{x}}} \to \Psi_i^{\bar{\mathrm{x}}}$ be defined inductively as follows: For every $S \in \mathcal{P}\mathrm{At}_{\bar{\mathrm{x}}}$ we set $\rho_n(S) := \exists x_n.\, \big(\bigwedge_{A \in S} A\big) \wedge \big(\bigwedge_{A \in \mathrm{At}_{\bar{\mathrm{x}}} \setminus S} \neg A\big)$. For every $i < n$ and every $S \in \mathcal{P}^{n-i+1}\mathrm{At}_{\bar{\mathrm{x}}}$ we set $\rho_i(S) := \exists x_i.\, \big(\bigwedge_{T \in S} \rho_{i+1}(T)\big) \wedge \bigwedge_{T \in \mathcal{P}^{n-i}\mathrm{At}_{\bar{\mathrm{x}}} \setminus S} \neg \rho_{i+1}(T)$. $\Psi_i^{\bar{\mathrm{x}}}$, $\widehat{\Psi}_i^{\bar{\mathrm{x}}}(\bar{r})$, ρ_i

Claim I: For every i and every $S \in \mathcal{P}^{n-i+1}\mathrm{At}_{\bar{\mathrm{x}}}$ we have $S \in \mu_{i-1}(\bar{r})$ if and only if $\rho_i(S) \in \widehat{\Psi}_i^{\bar{\mathrm{x}}}(\bar{r})$.

Proof: We proceed by induction on i, starting from $i = n$ going downwards.

Base case: Let $i = n$. Given any tuple $\bar{r} \in \mathbb{Q}^{n-1}$, for every set $S \in \mu_{n-1}(\bar{r})$ there is some $q_n \in \mathbb{Q}$ such that for every $A(\bar{\mathrm{x}}) \in \mathrm{At}_{\bar{\mathrm{x}}}$ we have $\mathbb{Q} \models A(\bar{r}, q_n)$ if and only if $A \in S$. In other words, we have

$$\mathbb{Q}, [x_1 \mapsto r_1, \ldots, x_{n-1} \mapsto r_{n-1}] \models \underbrace{\exists x_n.\, \Big(\bigwedge_{A(\bar{\mathrm{x}}) \in S} A(\bar{\mathrm{x}})\Big) \wedge \bigwedge_{A(\bar{\mathrm{x}}) \in \mathrm{At}_{\bar{\mathrm{x}}} \setminus S} \neg A(\bar{\mathrm{x}})}_{= \rho_n(S)} ,$$

which holds if and only if $\rho_n(S) \in \widehat{\Psi}_n^{\bar{\mathrm{x}}}(\bar{r})$.

Conversely, from $\rho_n(S) \in \widehat{\Psi}_n^{\bar{\mathrm{x}}}(\bar{r})$ follows the existence of some q_n for which $\mathbb{Q} \models A(\bar{r}, q_n)$ if and only if $A(\bar{\mathrm{x}}) \in S$. Then, $S \in \mu_{n-1}(\bar{r})$.

Inductive case: Let $i < n$. Given any tuple $\bar{r} \in \mathbb{Q}^{i-1}$, for every set $S \in \mu_{i-1}(\bar{r})$ there is some $q_i \in \mathbb{Q}$ such that $S = \mu_i(\bar{r}, q_i)$. By induction, for every $T \in \mathcal{P}^{n-(i+1)+1}\mathrm{At}_{\bar{\mathrm{x}}}$ we have $T \in S$ if and only if $\mathbb{Q}, [x_1 \mapsto r_1, \ldots, x_{i-1} \mapsto r_{i-1}, x_i \mapsto q_i] \models \rho_{i+1}(T)$. Hence, we have

$$\mathbb{Q}, [x_1 \mapsto r_1, \ldots, x_{i-1} \mapsto r_{i-1}] \models \underbrace{\exists x_i.\, \Big(\bigwedge_{T \in S} \rho_{i+1}(T)\Big) \wedge \bigwedge_{T \in (\mathcal{P}^{n-i}\mathrm{At}_{\bar{\mathrm{x}}}) \setminus S} \neg \rho_{i+1}(T)}_{= \rho_i(S)} ,$$

which holds if and only if $\rho_i(S) \in \widehat{\Psi}_i^{\bar{\mathrm{x}}}(\bar{r})$.

Conversely, from $\rho_i(S) \in \widehat{\Psi}_i^{\bar{\mathrm{x}}}(\bar{r})$ follows the existence of some q_i for which
$\mathbb{Q}, [x_1 \mapsto r_1, \ldots, x_{i-1} \mapsto r_{i-1}, x_i \mapsto q_i] \models \rho(T)$
if and only if $T \in S$. Then, $S \in \mu_{i-1}(\bar{r})$. ◊

The lemma follows by Lemma 7.1.5 and Claim I, which establishes the one-to-one correspondence between fingerprints in $\mu_{i-1}(\bar{r})$ and the formulas in $\widehat{\Psi}_i^{\bar{\mathrm{x}}}(\bar{r})$ for every i, $1 \leq i \leq n$, and every $\bar{r} \in \mathbb{Q}^{i-1}$. □

It follows by Lemmas 4.2.7 and 7.1.8 that, if the LRA sentence φ is valid in $\mathbb{Q}$, then there is a *μ-uniform satisfying strategy* σ (in a sense analogous to Definitions 4.2.2 and 4.2.6) whose *target set*

$\mathcal{T}_\sigma$

$$\mathcal{T}_\sigma := \bigcup_{1 \leq k \leq n} \big\{ q \in \mathbb{Q} \;\big|\; \text{there is some tuple } \bar{r} \in \mathbb{Q}^k \text{ such that } \sigma_k(\bar{r}) = q \big\}$$

contains at most

$$\begin{aligned}
\sum_{k=0}^{n} \prod_{j=0}^{k} \max_{\bar{r} \in \mathbb{Q}^j} \big(|\mu_j(\bar{r})|\big) \;&\leq\; (n+1) \cdot \big(c \cdot 2^{(d + \log|\mathrm{At}_{\bar{\mathrm{x}}}|) \cdot 2^n}\big)^{(n+1)} \\
&=\; 2^{\log(n+1)} \cdot 2^{(n+1) \cdot \log c} \cdot 2^{(n+1) \cdot (d + \log|\mathrm{At}_{\bar{\mathrm{x}}}|) \cdot 2^n} \\
&=\; 2^{\log(n+1) + (n+1) \cdot \log c + d \cdot (n+1) \cdot 2^n + (n+1) \cdot \log|\mathrm{At}_{\bar{\mathrm{x}}}| \cdot 2^n} \\
&\leq\; 2^{e \cdot n \cdot \log|\mathrm{At}_{\bar{\mathrm{x}}}| \cdot 2^n} \;.
\end{aligned}$$

elements for certain positive constants c, d, e. Notice that this is a slight improvement compared to the non-separated case, where the number of testpoints that need to be considered is in the worst case doubly exponential in $2n$ rather than n. Although this hardly changes the asymptotic behavior, it can make a difference in practice.

It is known that proving validity of a given LRA sentence under $\mathbb{Q}$ by means of virtual substitution requires worst-case computing time that is "polynomial in the length of the input formula, exponential in the number of quantified variables, and doubly exponential in the number of quantifier blocks" [LW93], page 1. Hence, the problem of checking validity of LRA sentences with a bounded number of quantifier alternations lies in ExpTime. When we consider any LRA sentence $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \psi$ with quantifier-free ψ in which universally and existentially quantified variables are separated, we obtain a similar result, if we bound the degree of interaction of existential and also universal variables by one. In other words, we require that the sets $\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n$ are all pairwise separated in φ. This separateness condition is even stricter than the one for the strongly separated fragment (cf. Definition 3.2.3). Then, by a straightforward adaptation of the proof of Lemma 3.2.5, we can argue that φ is equivalent to some LRA sentence φ' that is a Boolean combination of sentences of the form $\forall \bar{\mathrm{x}}_i . \chi_k(\bar{\mathrm{x}}_i)$ or $\exists \bar{\mathrm{y}}_i . \eta_\ell(\bar{\mathrm{y}}_i)$ for certain quantifier-free formulas χ_k and η_ℓ which exclusively contain atoms that stem from the original φ. Since there are only exponentially many such subformulas (up to equivalence), the length of φ' needs to be at most exponential in the length of φ. In order to decide whether φ' is satisfied under $\mathbb{Q}$, we can eliminate all quantifiers in the subsentences $\forall \bar{\mathrm{x}}_i . \chi_k(\bar{\mathrm{x}}_i)$ and $\exists \bar{\mathrm{y}}_i . \eta_\ell(\bar{\mathrm{y}}_i)$, treating each subsentence individually. Each such exhaustive elimination step can be done deterministically in at most exponential time in the length of the quantifier prefix and in polynomial time in the length of the subsentence. Therefore, deciding $\mathbb{Q} \models \varphi'$ lies in ExpTime and the same hold for deciding $\mathbb{Q} \models \varphi$.

Proposition 7.1.9. *Let BSF-LRA be the class of* block-separated LRA sentences *which we define to be the class of LRA sentences of the form $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \psi$ with quantifier-free ψ in which the sets $\bar{\mathrm{x}}_1, \ldots, \bar{\mathrm{x}}_n, \bar{\mathrm{y}}_1, \ldots, \bar{\mathrm{y}}_n$ are all pairwise separated. The problem of checking whether any given BSF-LRA sentence is satisfied under $\mathbb{Q}$ lies in* ExpTime.

7.2 Skolemization Policies Taking Weak Dependences into Account

Beyond the applicability to the classical decision problem, discovering that existential variables are
for proof complexity and automated reasoning. It is folklore in automated reasoning that shifting existential quantifiers inwards before applying Skolemization is beneficial in most cases. If there is potential for reducing the arity of Skolem functions, this potential may be used to reduce the length of shortest proofs or to substantially reduce the search space that theorem provers have to explore for finding proofs. The equivalences in Proposition 1.0.1 enable the formulation of certain strategies for quantifier shifting known as, e.g., *miniscoping* [NW01] or *anti-prenexing* [Egl94], that have turned out to be very useful in automated reasoning. Moreover, for first-order logic non-standard Skolemization techniques have been considered that lead to Skolem functions with smaller arity, see [BEL01, NW01] for an overview. In terms of proof complexity, choosing an unsuitable Skolemization technique can make shortest proofs exponentially or even non-elementarily longer in the worst case [Egl94, BL94, BFL94].

Remark 7.2.1. *In the fields of QBF solving and CSP*[2] *solving so-called* dependency schemes *have been developed that help to exploit Boolean structure to optimize the arity of Skolem functions [Sam08, SS09, Lon12, BB16]. For QBF sentence it is clear that an analysis based on separateness of (propositional) variables yields only trivial results, as every atom can contain at most one such variable. However, the fact that QBF solvers strongly benefit from the optimization techniques based on dependency schemes, one could hope for a significant impact in first-order reasoning as well. It seems to be promising to lift QBF dependency schemes to first-order logic and then combine them with analysis techniques based on separateness — see also Section 3.6 for the combined analysis of separateness and Boolean structure.*

There has been work on exploiting certain forms of independence in first-order logic as well. Notably, any work that tries to improve the results of Skolemization does, at least implicitly, long for the minimization of dependencies between existentially and universally quantified variables. Examples of improved Skolemization policies can be found in [BEL01, NW01]. Goubault-Larrecq [Gou95] has analyzed dependencies in first-order logic, using first-order BDDs as a tool.

To the best of the present author's knowledge, currently used techniques in automated reasoning do not detect and exploit weak dependences, as defined and investigated in Chapter 4. On the syntactic level, we introduced Lemma 3.2.4 in Section 3.2 as a tool for shifting existential quantifiers $\exists y$ also into certain conjunctions although the bound variable y may occur in each conjunct. The price that has to be paid is a potentially exponential blowup of the formula. What we possibly gain are smaller arities of Skolem functions. In the extreme case of GBSR sentences, we even get Skolem constants in the end, which, in terms of small arities of Skolem functions, is clearly the best result we could possibly achieve. On the other hand, reducing the arity of Skolem functions comes at the price of an increasing number of Skolem functions (with smaller arity). This trade-off situation — formula length vs. arities of Skolem functions vs. number of Skolem functions — has to be dealt with gracefully. It seems to be an interesting direction for future research to find out when exactly automated reasoning benefits from quantifier shifting in the spirit of Lemma 3.2.4 and when the additional cost outweighs the potential gain. This is even more interesting as the described ideas are not limited in scope to pure uninterpreted first-order logic. Some of the employed proof methods are completely oblivious to predicate symbols or function symbols with a-priori-fixed interpretations.

Example 7.2.2. *Consider the open formula*

$$\varphi := \forall z \forall x \exists y. \big(P(x) \vee R(y,z)\big) \wedge Q(y,y).$$

Standard quantifier shifting cannot move the $\exists y$ inwards any further. Naive Skolemization yields

$$\varphi' := \forall z \forall x. \big(P(x) \vee R(f_y(z,x),z)\big) \wedge Q(f_y(z,x), f_y(z,x)).$$

[2] QBF stands for *quantified Boolean formulas*; CSP stands for *constraint satisfaction problems.*

On the other hand, applying a simplified variant of Lemma 3.2.4 to φ and removing redundant subformulas results in

$$\forall z \forall x. \big(\exists y_2.\, Q(y_2, y_2)\big) \wedge \big(P(x) \vee \exists y_3. \big(R(y_3, z) \wedge Q(y_3, y_3)\big)\big).$$

Since universal quantifiers distribute over conjunctions, we can now apply standard quantifier shifting to move the quantifiers $\forall z \forall x$ inwards and obtain

$$\big(\exists y_2.\, Q(y_2, y_2)\big) \wedge \Big(\big(\forall x.\, P(x)\big) \vee \big(\forall z \exists y_3.\, R(y_3, z) \wedge Q(y_3, y_3)\big)\Big).$$

Skolemization of this result leads to

$$\varphi'' := \big(Q(c_{y_2}, c_{y_2})\big) \wedge \Big(\big(\forall x.\, P(x)\big) \vee \big(\forall z.\, R(f_{y_3}(z), z) \wedge Q(f_{y_3}(z), f_{y_3}(z))\big)\Big).$$

Comparing φ' and φ'', we see that we trade a sentence with a single Skolem term $f_y(z, x)$ and three distinct atoms for a sentence with two Skolem terms c_{y_2} and $f_{y_3}(z)$ and four distinct atoms.

The following considerations constitute a first step towards Skolemization methods that are sensitive to weak dependences. But before we start, one more remark is in order concerning the relation to known concepts of quantification that include explicit dependence information.

Remark 7.2.3. *In [Hen61] Henkin introduced a generalized form of existential quantifiers, to which we shall refer as* Henkin quantifiers*, but they are sometimes also called* finite partially ordered quantifiers *or* branching quantifiers *or* nonlinear quantifiers *(see [KM95] for a broader overview.) Henkin quantifiers can explicitly express dependence of existentially quantified variables on universally quantified ones. For instance, in the sentence $\psi := \forall z \forall x \exists_z y.\, Q(z, y) \leftrightarrow P(x)$ the value of y may depend on the value of z but has to be independent from x's value. This sentence is equivalent to the second-order formula $\psi_{Sk} := \exists f_y. \forall z \forall x.\, Q(z, f_y(z)) \leftrightarrow P(x)$ for some Skolem function f_y. One could say that Henkin quantifiers make* in*dependence explicit by* not *listing certain universally quantified variables as a subscript.*

The patterns of weak vs. strong dependence induced by separateness of variables are more subtle than a strict classification into full dependence vs. full independence as encouraged by Henkin-style quantification. For example, the slightly modified sentence $\varphi := \forall z \forall x \exists y.\, Q(z, y) \leftrightarrow P(x)$ is equivalent to

$$\varphi' := \forall z \exists y_1 y_2 \forall x. \big(Q(z, y_1) \rightarrow P(x)\big) \wedge \big(P(x) \rightarrow Q(z, y_2)\big),$$

where the weakness of the dependence of y on x in φ becomes evident. Using second-order quantifiers, we can make this explicit:

$$\varphi'' := \exists g_1 g_2. \forall x. \bigvee\nolimits_{i \in \{1,2\}} \forall z. \big(Q(z, g_i(z)) \leftrightarrow P(x)\big),$$

which is again equivalent to φ.

Altogether, the example illustrates that separateness of existentially quantified and universally quantified variables leads to a certain degree of independence, but it does not reach the level of independence Henkin quantifiers can guarantee. This is not at all surprising, because Henkin quantifiers increase the expressiveness of first-order logic significantly.

In recent years concepts of dependence and independence in first-order logic have been studied that are much more sophisticated than Henking quantifiers. This has recently become an active field of research. Introductory material and further references can be found in [Vää07, GKVV16, AKVV16]. Possible relations to the notion of weak dependence studied in Chapter 4 and the present section remain to be investigated.[3]

The transformations outlined in Example 7.2.2 have been done more systematically in the proof of Lemma 2.0.3 and 2.0.4 (Chapter 2). Moreover, this approach was central to the numerous translation procedures devised in Chapter 3, e.g. for translating GBSR into BSR, GAF into AF, and so on. Schematically, the technique can be stated as follows:

basic formulas

Consider a relational first-order sentence $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n.\, \psi$ in standard form. A *basic formula* is any atom A, any negated atom $\neg A$, and any formula $\mathcal{Q}v.\, \chi$ with $\mathcal{Q} \in \{\exists, \forall\}$. An occurrence of a basic formula η in a formula ψ is considered *maximal*, if the occurrence is not a proper part of an occurrence of another basic formula. We start by setting $k := 0$ and $\psi_0 := \psi$:

[3] This was suggested to the author of the present thesis by Erich Grädel at the Algorithmic Model Theory Meeting in Berlin, Germany, in March 2018.

(1) Transform ψ_k into a disjunction of conjunctions of basic formulas (maximal occurrences of basic formulas in ψ_k are treated as indivisible units) by application of the laws of Boolean algebra.

(2) Set $\psi'_k := \exists \bar{\mathrm{y}}_{n-k}.\, \psi_k$ and shift the leading existential quantifiers inwards as far as possible. We do not rename bound variables and thus the result may contain multiple occurrences of quantifiers with the same variable name.

(3) Transform ψ'_k into a conjunction of disjunctions of basic formulas (maximal occurrences of basic formulas in ψ'_k are treated as indivisible units) by application of the laws of Boolean algebra.

(4) Set $\psi_{k+1} := \forall \bar{\mathrm{x}}_{n-k}.\, \psi'_k$ and shift the leading universal quantifiers inwards as far as possible. Again, we do not rename bound variables.

(5) Stop if $k = n - 1$, otherwise increment k by one and continue at Step (1).

In the previous sections we have often assumed that certain sets of variables are separated and have then shown that after applying the above scheme there is no quantifier $\mathcal{Q}v$ whose scope contains variables from two separated sets. We now proceed in the other direction and devise an overapproximation of the variables captured in the scopes of quantifiers in ψ_n. The analysis will be based on φ and we will be using reasonably simple syntactic criteria.

Let $\bar{\mathrm{x}} := \bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_n$ and $\bar{\mathrm{y}} := \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_n$. Moreover, let At be the set of all atoms occurring in φ. In this section, we need a refinement of the notion of the index of a variable.

Definition 7.2.4 (Extended upward closure of captured quantified variables)**.** *The* block index *of a variable $v \in \bar{\mathrm{x}} \cup \bar{\mathrm{y}}$ is denoted by* $\mathrm{bidx}_\varphi(v)$ *and defined such that* $\mathrm{bidx}_\varphi(v)$

$$\mathrm{bidx}_\varphi(v) := \begin{cases} 2k-1 & \text{if } v \in \bar{\mathrm{x}}_k, \\ 2k & \text{if } v \in \bar{\mathrm{y}}_k. \end{cases}$$

Let $\preceq_\varphi$ be the smallest reflexive and transitive relation over the variables in φ such that $v \preceq_\varphi v'$ whenever $\mathrm{bidx}_\varphi(v) \leq \mathrm{bidx}_\varphi(v')$ and there is some atom in φ in which v and v' co-occur. For every variable v in φ the upward closure $V_\varphi^{\succeq v}$ *is the set $\{v' \in \mathrm{vars}(\varphi) \mid v \preceq_\varphi v'\}$. The* extended upward closure $\widehat{V}_\varphi^{\succeq v}$ *is the smallest set satisfying the following properties:* $v \preceq_\varphi v'$ $V_\varphi^{\succeq v}$, $\widehat{V}_\varphi^{\succeq v}$

(a) $V_\varphi^{\succeq v} \subseteq \widehat{V}_\varphi^{\succeq v}$, and

(b) for all v', v'' with $\mathrm{bidx}_\varphi(v) \leq \mathrm{bidx}_\varphi(v') \leq \mathrm{bidx}_\varphi(v'')$ and $v'' \in \widehat{V}_\varphi^{\succeq v} \cap V_\varphi^{\succeq v'}$ we have $V_\varphi^{\succeq v'} \subseteq \widehat{V}_\varphi^{\succeq v}$.

For all of the above notations we omit the subscript φ if the respective φ is clear from the context.

Intuitively, for any quantifier $\mathcal{Q}v$ in φ the set $\widehat{V}^{\prec v}$ overapproxiamtes the set of all bound variables that occur in the scope of $\mathcal{Q}v$ during and after finishing the procedure described above. We leave it to the reader to prove this intuition formally.

Lemma 7.2.5. *Let $u, v \in \mathrm{vars}(\varphi)$ be any two variables in φ.*

(i) If $\mathrm{bidx}(u) = \mathrm{bidx}(v)$, then we either have $\widehat{V}^{\succeq u} = \widehat{V}^{\succeq v}$ or $\widehat{V}^{\succeq u} \cap \widehat{V}^{\succeq v} = \emptyset$.

(ii) If $\mathrm{bidx}(u) < \mathrm{bidx}(v)$, then we either have $\widehat{V}^{\succeq v} \subsetneq \widehat{V}^{\succeq u}$ or $\widehat{V}^{\succeq v} \cap \widehat{V}^{\succeq u} = \emptyset$.

Proof. We start by proving the following auxiliary results:

Claim I: Let w, z be two variables in φ with $\mathrm{bidx}(w) \geq \mathrm{bidx}(z)$. Then, $V^{\succeq w} \subseteq \widehat{V}^{\succeq z}$ entails $\widehat{V}^{\succeq w} \subseteq \widehat{V}^{\succeq z}$.

Proof: For every $w'' \in \widehat{V}^{\trianglerighteq w}$ there is some chain of variables $w'_1, w''_1, \ldots, w'_m, w''_m$ such that

$$\begin{array}{l} \mathrm{bidx}(w) \leq \mathrm{bidx}(w'_i) \leq \mathrm{bidx}(w''_i) \text{ for every } i, \text{ and} \\ w''_1 \in V^{\trianglerighteq w} \cap V^{\trianglerighteq w'_1}, \\ \quad w''_2 \in V^{\trianglerighteq w'_1} \cap V^{\trianglerighteq w'_2}, \\ \quad \vdots \\ \quad w''_m \in V^{\trianglerighteq w'_{m-1}} \cap V^{\trianglerighteq w'_m}, \\ \quad w'' \in V^{\trianglerighteq w'_m}. \end{array}$$

By definition of $\widehat{V}^{\trianglerighteq z}$, our assumption $V^{\trianglerighteq w} \subseteq \widehat{V}^{\trianglerighteq z}$ entails that the existence of the chain $w'_1, w''_1, \ldots, w'_m, w''_m$ leads to $V^{\trianglerighteq w'_i} \subseteq \widehat{V}^{\trianglerighteq z}$ for every w'_i. Hence, $w'' \in \widehat{V}^{\trianglerighteq z}$. In other words, we have $\widehat{V}^{\trianglerighteq w} \subseteq \widehat{V}^{\trianglerighteq z}$. ◊

Claim II: Let w, z be two variables in φ with $\mathrm{bidx}(w) \geq \mathrm{bidx}(z)$. If there is some variable x'' with $x'' \in \widehat{V}^{\trianglerighteq w} \cap \widehat{V}^{\trianglerighteq z}$, then $\widehat{V}^{\trianglerighteq w} \subseteq \widehat{V}^{\trianglerighteq z}$.

Proof: Because of $x'' \in \widehat{V}^{\trianglerighteq w}$ we have $\mathrm{bidx}(x'') \geq \mathrm{bidx}(w)$ and there is some chain of variables $w'_1, w''_1, \ldots, w'_m, w''_m$ such that

$$\begin{array}{l} \mathrm{bidx}(z) \leq \mathrm{bidx}(w) \leq \mathrm{bidx}(w'_i) \leq \mathrm{bidx}(w''_i) \text{ for every } i, \text{ and} \\ w''_1 \in V^{\trianglerighteq w} \cap V^{\trianglerighteq w'_1}, \\ \quad w''_2 \in V^{\trianglerighteq w'_1} \cap V^{\trianglerighteq w'_2}, \\ \quad \vdots \\ \quad w''_m \in V^{\trianglerighteq w'_{m-1}} \cap V^{\trianglerighteq w'_m}, \\ \quad x'' \in V^{\trianglerighteq w'_m}. \end{array}$$

Because of $x'' \in \widehat{V}^{\trianglerighteq z}$ we have $\mathrm{bidx}(x'') \geq \mathrm{bidx}(z)$ and there is some chain of variables $z'_1, z''_1, \ldots, z'_q, z''_q$ such that

$$\begin{array}{l} \mathrm{bidx}(z) \leq \mathrm{bidx}(z'_j) \leq \mathrm{bidx}(z''_j) \text{ for every } j, \text{ and} \\ z''_1 \in V^{\trianglerighteq z} \cap V^{\trianglerighteq z'_1}, \\ \quad z''_2 \in V^{\trianglerighteq z'_1} \cap V^{\trianglerighteq z'_2}, \\ \quad \vdots \\ \quad z''_q \in V^{\trianglerighteq z'_{q-1}} \cap V^{\trianglerighteq z'_q}, \\ \quad x'' \in V^{\trianglerighteq z'_q}. \end{array}$$

By definition of $\widehat{V}^{\trianglerighteq z}$, we get $V^{\trianglerighteq z'_j} \subseteq \widehat{V}^{\trianglerighteq z}$ for every z'_j (following the chain $V^{\trianglerighteq z} V^{\trianglerighteq z'_1} \ldots V^{\trianglerighteq z'_q}$) and $V^{\trianglerighteq w'_i} \subseteq \widehat{V}^{\trianglerighteq z}$ for every w'_i (following the chain $V^{\trianglerighteq w'_m} \ldots V^{\trianglerighteq w'_1} V^{\trianglerighteq w}$) and $V^{\trianglerighteq w} \subseteq \widehat{V}^{\trianglerighteq z}$. By virtue of Claim I, the last observation entails $\widehat{V}^{\trianglerighteq w} \subseteq \widehat{V}^{\trianglerighteq z}$. ◊

Assume $\mathrm{bidx}(u) = \mathrm{bidx}(v)$ and suppose that $\widehat{V}^{\trianglerighteq u} \cap \widehat{V}^{\trianglerighteq v}$ contains at least one variable w. By virtue of Claim II, we get $\widehat{V}^{\trianglerighteq u} \subseteq \widehat{V}^{\trianglerighteq v}$ and $\widehat{V}^{\trianglerighteq v} \subseteq \widehat{V}^{\trianglerighteq u}$. Therefore, $\widehat{V}^{\trianglerighteq u} = \widehat{V}^{\trianglerighteq v}$.

Now assume $\mathrm{bidx}(u) < \mathrm{bidx}(v)$ and suppose that $\widehat{V}^{\trianglerighteq u} \cap \widehat{V}^{\trianglerighteq v}$ contains at least one variable w. By virtue of Claim II, we get $\widehat{V}^{\trianglerighteq v} \subseteq \widehat{V}^{\trianglerighteq u}$. Moreover, we have $u \in V^{\trianglerighteq u} \subseteq \widehat{V}^{\trianglerighteq u}$ on the one hand but $u \notin \widehat{V}^{\trianglerighteq v}$ on the other hand. The latter holds as for every $z \in \widehat{V}^{\trianglerighteq v}$ we have $\mathrm{bidx}(z) \geq \mathrm{bidx}(v)$. Consequently, $\widehat{V}^{\trianglerighteq u} \supsetneq \widehat{V}^{\trianglerighteq v}$. □

$\mathrm{At}_\varphi(V)$

For any set $V \subseteq \mathrm{vars}(\varphi)$ we denote by $\mathrm{At}_\varphi(V)$ the set of all atoms occurring in φ that contain at least one variable from V. The following lemma follows immediately from Lemma 7.2.5.

Lemma 7.2.6. *Let $u, v \in \mathrm{vars}(\varphi)$ be two variables for which $\mathrm{bidx}(u) \leq \mathrm{bidx}(v)$ and $v \notin \widehat{V}^{\trianglerighteq u}$. Then, $\mathrm{At}_\varphi(\widehat{V}^{\trianglerighteq u}) \cap \mathrm{At}_\varphi(\widehat{V}^{\trianglerighteq v}) = \emptyset$ and, hence, the sets $\widehat{V}^{\trianglerighteq u}$ and $\widehat{V}^{\trianglerighteq v}$ are separated in φ.*

It is now our aim to show that any existentially quantified variable y_* in φ depends only weakly on any universally quantified v if there is no z such that $\widehat{V}^{\preceq z}$ contains both y_* and v. Moreover, we intend to exploit this insight when Skolemizing any of the existentially quantified variables in φ. We start with a syntactic approach. A semantically-minded alternative will follow later.

Theorem 7.2.7. *Let $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \psi$ be a first-order sentence in standard form with quantifier-free ψ. Consider any k, $1 \leq k \leq n$, and let y_* be some variable in $\bar{\mathrm{y}}_k$. Let W be the union of all sets $\widehat{V}_{\varphi}^{\preceq v}$ that contain y_* and let $\bar{\mathrm{x}}_*$ be a tuple containing exactly the variables x from $(\bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_k) \cap W$. Then, there is some positive integer m such that φ is equivalent to some sentence of the form* W

$$\varphi' := \exists f_1 \ldots f_m . \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_k \exists \big(\bar{\mathrm{y}}_k \setminus \{y_*\}\big) . \bigvee_{1 \leq j \leq m} \psi' \big[y_* / f_j(\bar{\mathrm{x}}_*)\big] \ .$$

Moreover, every atom in ψ' is a copy of some atom from ψ where bound variables are possibly renamed after shifting quantifiers.

Proof sketch. Let $\overline{W}$ be the complement of W, i.e. $\overline{W} := \mathrm{vars}(\varphi) \setminus W$. By Lemmas 7.2.5 and 7.2.6, $\overline{W}$ we observe the following properties for all $u \in W$ and $v \in \overline{W}$:

(a) $\widehat{V}_{\varphi}^{\preceq u} \cap \widehat{V}_{\varphi}^{\preceq v} = \emptyset$,

(b) $\widehat{V}_{\varphi}^{\preceq u} \subseteq W$ and $\widehat{V}_{\varphi}^{\preceq v} \subseteq \overline{W}$, and

(c) the sets $\widehat{V}_{\varphi}^{\preceq u}, \widehat{V}_{\varphi}^{\preceq v}$ are separated in φ.

It follows that the sets W and $\overline{W}$ are separated in φ.

Let $\bar{\mathrm{u}}^i, \bar{\mathrm{v}}^i, \bar{\mathrm{w}}^i, \bar{\mathrm{z}}^i$ with $i = 1, 2$ be tuples of variables defined as follows:

$\bar{\mathrm{u}}^1$ contains all the variables from $\overline{W} \cap (\bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_k)$,

$\bar{\mathrm{u}}^2$ contains all the variables from $\overline{W} \cap (\bar{\mathrm{x}}_{k+1} \cup \ldots \cup \bar{\mathrm{x}}_n)$,

$\bar{\mathrm{v}}^1$ contains all the variables from $\overline{W} \cap (\bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_k)$,

$\bar{\mathrm{v}}^2$ contains all the variables from $\overline{W} \cap (\bar{\mathrm{y}}_{k+1} \cup \ldots \cup \bar{\mathrm{y}}_n)$,

$\bar{\mathrm{w}}^1$ contains all the variables from $W \cap (\bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_k)$,

$\bar{\mathrm{w}}^2$ contains all the variables from $W \cap (\bar{\mathrm{x}}_{k+1} \cup \ldots \cup \bar{\mathrm{x}}_n)$,

$\bar{\mathrm{z}}^1$ contains all the variables from $W \cap (\bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_k) \setminus \{y_*\}$,

$\bar{\mathrm{z}}^2$ contains all the variables from $W \cap (\bar{\mathrm{y}}_{k+1} \cup \ldots \cup \bar{\mathrm{y}}_n)$.

Then, the sets $\bar{\mathrm{u}}^1 \cup \bar{\mathrm{u}}^2 \cup \bar{\mathrm{v}}^1 \cup \bar{\mathrm{v}}^2$ and $\bar{\mathrm{w}}^1 \cup \bar{\mathrm{w}}^2 \cup \{y_*\} \cup \bar{\mathrm{z}}^1 \cup \bar{\mathrm{z}}^2$ are separated in φ. Using the techniques from the proofs of Lemmas 2.0.3 and 2.0.4, we can transform φ into an equivalent sentence φ'' of φ'' the form

$$\varphi'' := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_k \exists \big(\bar{\mathrm{y}}_k \setminus \{y_*\}\big) \exists y_* . \bigvee_{i \in I} \chi_i(\bar{\mathrm{u}}^1, \bar{\mathrm{v}}^1) \wedge \eta_i(\bar{\mathrm{w}}^1, y_*, \bar{\mathrm{z}}^1) \ ,$$

where I is some finite set of indices, the quantifiers for the variables in $\bar{\mathrm{u}}^2 \cup \bar{\mathrm{v}}^2$ have been shifted into the subformulas $\chi_i(\bar{\mathrm{u}}^1, \bar{\mathrm{v}}^1)$, and the quantifiers for the variables in $\bar{\mathrm{w}}^2 \cup \bar{\mathrm{z}}^2$ have been shifted into the subformulas $\eta_i(\bar{\mathrm{w}}^1, y_*, \bar{\mathrm{z}}^1)$. Next, we show that φ'' is equivalent to some sentence φ' that φ' has the form

$$\varphi' := \exists f_1 \ldots f_m . \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_k \exists (\bar{\mathrm{y}}_k \setminus \{y_*\}) . \bigvee_{1 \leq k \leq m} \bigvee_{i \in I} \chi_i(\bar{\mathrm{u}}^1, \bar{\mathrm{v}}^1) \wedge \eta_i(\bar{\mathrm{w}}^1, y_*, \bar{\mathrm{z}}^1) \big[y_* / f_k(\bar{\mathrm{w}}^1)\big] \ .$$

In the rest of the proof, we use $\bar{\mathrm{u}}^1, \bar{\mathrm{v}}^1, \bar{\mathrm{w}}^1, \bar{\mathrm{z}}^1$ without superscripts, i.e. we write $\bar{\mathrm{u}}, \bar{\mathrm{v}}, \bar{\mathrm{w}}, \bar{\mathrm{z}}$ instead.

Any model of φ' is certainly a model of φ'', i.e. we have $\varphi' \models \varphi''$. Let $\mathcal{A}$ be any model of φ''. Then, there are mappings $\sigma_{\bar{\mathsf{v}}} : \mathsf{A}^{|\bar{\mathsf{u}}|} \times \mathsf{A}^{|\bar{\mathsf{w}}|} \to \mathsf{A}^{|\bar{\mathsf{v}}|}$, $\sigma_{y_*} : \mathsf{A}^{|\bar{\mathsf{u}}|} \times \mathsf{A}^{|\bar{\mathsf{w}}|} \to \mathsf{A}$, and $\sigma_{\bar{\mathsf{z}}} : \mathsf{A}^{|\bar{\mathsf{u}}|} \times \mathsf{A}^{|\bar{\mathsf{w}}|} \to \mathsf{A}^{|\bar{\mathsf{z}}|}$ that resemble parts of some satisfying strategy that exists for φ''. More precisely, we have

$$\mathcal{A} \models \bigvee_{i \in I} \chi_i\big(\bar{\mathsf{a}}, \sigma_{\bar{\mathsf{v}}}(\bar{\mathsf{a}}, \bar{\mathsf{c}})\big) \wedge \eta_i\big(\bar{\mathsf{c}}, \sigma_{y_*}(\bar{\mathsf{a}}, \bar{\mathsf{c}}), \sigma_{\bar{\mathsf{z}}}(\bar{\mathsf{a}}, \bar{\mathsf{c}})\big)$$

for all $\bar{\mathsf{a}} \in \mathsf{A}^{|\bar{\mathsf{u}}|}$ and $\bar{\mathsf{c}} \in \mathsf{A}^{|\bar{\mathsf{w}}|}$. Every pair $\bar{\mathsf{a}}, \bar{\mathsf{b}}$ of tuples $\bar{\mathsf{a}} \in \mathsf{A}^{|\bar{\mathsf{u}}|}$ and $\bar{\mathsf{b}} \in \mathsf{A}^{|\bar{\mathsf{v}}|}$ can be characterized by the set $S_{\bar{\mathsf{a}},\bar{\mathsf{b}}} := \big\{i \in I \bigm| \mathcal{A} \models \chi_i(\bar{\mathsf{a}}, \bar{\mathsf{b}})\big\}$. Similarly, for every pair $\bar{\mathsf{c}}, \bar{\mathsf{d}}$ with $\bar{\mathsf{c}} \in \mathsf{A}^{|\bar{\mathsf{w}}|}$ and $\bar{\mathsf{d}} \in \mathsf{A}^{|\bar{\mathsf{z}}|}$ we define the set $T_{\bar{\mathsf{c}},\bar{\mathsf{d}}} := \big\{i \in I \bigm| \mathcal{A} \models \exists y_*.\, \eta_i\big(\bar{\mathsf{c}}, y_*, \bar{\mathsf{d}}\big)\big\}$. Then, we observe that $\mathcal{A} \models \bigvee_{i \in I} \chi_i\big(\bar{\mathsf{a}}, \sigma_{\bar{\mathsf{v}}}(\bar{\mathsf{a}}, \bar{\mathsf{c}})\big) \wedge \exists y_*.\, \eta_i\big(\bar{\mathsf{c}}, y_*, \sigma_{\bar{\mathsf{z}}}(\bar{\mathsf{a}}, \bar{\mathsf{c}})\big)$ holds if and only if $S_{\bar{\mathsf{a}},\sigma_{\bar{\mathsf{v}}}(\bar{\mathsf{a}},\bar{\mathsf{c}})} \cap T_{\bar{\mathsf{c}},\sigma_{\bar{\mathsf{z}}}(\bar{\mathsf{a}},\bar{\mathsf{c}})}$ is not empty.

$S_{\bar{\mathsf{a}},\bar{\mathsf{b}}}$, $T_{\bar{\mathsf{c}},\bar{\mathsf{d}}}$

For every $\bar{\mathsf{c}}$ there is a smallest integer $m_{\bar{\mathsf{c}}}$ and two lists $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{m_{\bar{\mathsf{c}}}}$ and $T_1, \ldots, T_{m_{\bar{\mathsf{c}}}}$ of tuples $\bar{\mathsf{a}}_k \in \mathsf{A}^{|\bar{\mathsf{u}}|}$ and of sets $T_k \subseteq I$ that satisfy the following properties.

(a) For every k we have $T_k = T_{\bar{\mathsf{c}},\sigma_{\bar{\mathsf{z}}}(\bar{\mathsf{a}}_k,\bar{\mathsf{c}})}$.

(b) For every $\bar{\mathsf{a}}$ the set $S_{\bar{\mathsf{a}},\sigma_{\bar{\mathsf{v}}}(\bar{\mathsf{a}},\bar{\mathsf{c}})}$ shares at least one element with at least one of the T_k.

Hence, there exist mappings $\tau_{\bar{\mathsf{z}}} : \mathsf{A}^{|\bar{\mathsf{u}}|} \times \mathsf{A}^{|\bar{\mathsf{w}}|} \to \mathsf{A}^{|\bar{\mathsf{z}}|}$ and $\tau_{y_*} : \mathsf{A}^{|\bar{\mathsf{u}}|} \times \mathsf{A}^{|\bar{\mathsf{w}}|} \to \mathsf{A}$ such that for every $\bar{\mathsf{a}}$

(i) there is some k such that $\tau_{\bar{\mathsf{z}}}(\bar{\mathsf{a}}, \bar{\mathsf{c}}) = \sigma_{\bar{\mathsf{z}}}(\bar{\mathsf{a}}_k, \bar{\mathsf{c}})$,

(ii) $T_{\bar{\mathsf{c}},\tau_{\bar{\mathsf{z}}}(\bar{\mathsf{a}},\bar{\mathsf{c}})} \cap S_{\bar{\mathsf{a}},\sigma_{\bar{\mathsf{v}}}(\bar{\mathsf{a}},\bar{\mathsf{c}})}$ contains at least one index j, and

(iii) τ_{y_*} is defined such that $\mathcal{A} \models \eta_j\big(\bar{\mathsf{c}}, \tau_{y_*}(\bar{\mathsf{a}}, \bar{\mathsf{c}}), \tau_{\bar{\mathsf{z}}}(\bar{\mathsf{a}}, \bar{\mathsf{c}})\big)$ for some $j \in S_{\bar{\mathsf{a}},\sigma_{\bar{\mathsf{v}}}(\bar{\mathsf{a}},\bar{\mathsf{c}})}$.

This entails $\mathcal{A} \models \bigvee_{i \in I} \chi_i\big(\bar{\mathsf{a}}, \sigma_{\bar{\mathsf{v}}}(\bar{\mathsf{a}}, \bar{\mathsf{c}})\big) \wedge \eta_i\big(\bar{\mathsf{c}}, \tau_{y_*}(\bar{\mathsf{a}}, \bar{\mathsf{c}}), \tau_{\bar{\mathsf{z}}}(\bar{\mathsf{a}}, \bar{\mathsf{c}})\big)$.

Let $m := \max\{m_{\bar{\mathsf{c}}} \mid \bar{\mathsf{c}} \in \mathsf{A}^{|\bar{\mathsf{w}}|}\}$. Notice that $m \leq |I|$. Let $\tau_1, \ldots, \tau_m$ be mappings with the signature $\tau_k : \mathsf{A}^{|\bar{\mathsf{w}}|} \to \mathsf{A}$ such that for every $\bar{\mathsf{c}}$ and every k, $1 \leq k \leq m_{\bar{\mathsf{c}}}$, we set $\tau_k(\bar{\mathsf{c}}) := \tau_{y_*}(\bar{\mathsf{a}}_k, \bar{\mathsf{c}})$ — for $k > m_{\bar{\mathsf{c}}}$ we can define $\tau_k(\bar{\mathsf{c}})$ arbitrarily. Then, we observe

$$\begin{aligned}\mathcal{A} &\models \bigvee_{i \in I} \Big(\chi_i\big(\bar{\mathsf{a}}, \sigma_{\bar{\mathsf{v}}}(\bar{\mathsf{a}}, \bar{\mathsf{c}})\big) \wedge \bigvee_{1 \leq k \leq m} \eta_i\big(\bar{\mathsf{c}}, \tau_k(\bar{\mathsf{c}}), \tau_{\bar{\mathsf{z}}}(\bar{\mathsf{a}}, \bar{\mathsf{c}})\big)\Big) \\ &\models\!\!\models \bigvee_{1 \leq k \leq m} \bigvee_{i \in I} \chi_i\big(\bar{\mathsf{a}}, \sigma_{\bar{\mathsf{v}}}(\bar{\mathsf{a}}, \bar{\mathsf{c}})\big) \wedge \eta_i\big(\bar{\mathsf{c}}, \tau_k(\bar{\mathsf{c}}), \tau_{\bar{\mathsf{z}}}(\bar{\mathsf{a}}, \bar{\mathsf{c}})\big)\ .\end{aligned}$$

This entails that $\mathcal{A}$ satisfies φ', which finishes the proof. □

Notice that we used a quite different notion of fingerprint in the proof of Theorem 7.2.7 than we have used before. This approach is encouraged by the special syntactic form of the matrix of φ'.

Theorem 7.2.7 is somewhat unsatisfactory, because the shape of the formula ψ' can only be determined by following the scheme described earlier (page 192), which in general requires a lot of syntactic transformations. It seems to be much more appealing to find a purely semantic argument (or a way of gracefully undoing the syntactic transformations), which in the end gives us a non-standard Skolemization technique that requires only straightforward replacement of subformulas. In the rest of the present section we shall make such an attempt and show the following result.

Theorem 7.2.8. *Let $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n.\, \psi$ be a first-order sentence in standard form with quantifier-free ψ. Consider any k, $1 \leq k \leq n$, and let y_* be some variable in $\bar{\mathrm{y}}_k$. Let W be the union of all sets $\widehat{V}_{\varphi}^{\preceq v}$ that contain y_* and let $\bar{\mathrm{x}}_*$ be a tuple containing exactly the variables x from $(\bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_k) \cap W$. Then, there is some positive integer m such that φ is equivalent to the sentence*

W

$$\varphi' := \exists f_1 \ldots f_m.\, \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_k \exists (\bar{\mathrm{y}}_k \setminus \{y_*\}).\ \bigvee_{1 \leq i \leq m} \forall \bar{\mathrm{x}}_{k+1} \exists \bar{\mathrm{x}}_{k+1} \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n.\, \psi\big[y_*/f_i(\bar{\mathrm{x}}_*)\big]\ .$$

Fix any structure $\mathcal{A}$ over the vocabulary of φ and consider the model-checking game associated with φ over $\mathcal{A}$. In the context of Theorem 7.2.8, we use the notion of *strategy* and related concepts from Chapter 4. Again, let $\overline{W} := \mathrm{vars}(\varphi) \setminus W$ denote the complement of W. By Lemmas 7.2.5 and 7.2.6, we know that W and $\overline{W}$ are separated in φ and that, hence, the sets of atoms $\mathrm{At}(W)$ and $\mathrm{At}(\overline{W})$ are disjoint (see also the proof of Theorem 7.2.7). Recall that a strategy σ is a sequence $\langle \sigma_1, \ldots, \sigma_n \rangle$ of mappings $\sigma_i : \mathsf{A}^{|\bar{\mathrm{x}}_1|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{x}}_i|} \to \mathsf{A}^{|\bar{\mathrm{y}}_i|}$. We divide every $\bar{\mathrm{x}}_i$ into two parts $\bar{\mathrm{x}}_i^1, \bar{\mathrm{x}}_i^2$ such that $\bar{\mathrm{x}}_i^1 = \bar{\mathrm{x}}_i \cap \overline{W}$ and $\bar{\mathrm{x}}_i^2 = \bar{\mathrm{x}}_i \cap W$. Analogously, we divide every $\bar{\mathrm{y}}_i$ into two parts $\bar{\mathrm{y}}_i^1$, $\bar{\mathrm{y}}_i^2$. For convenience, we also split every σ_i into two parts: $\sigma_i^1 : \mathsf{A}^{|\bar{\mathrm{x}}_1^1|} \times \mathsf{A}^{|\bar{\mathrm{x}}_1^2|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{x}}_i^1|} \times \mathsf{A}^{|\bar{\mathrm{x}}_i^2|} \to \mathsf{A}^{|\bar{\mathrm{y}}_i^1|}$ and $\sigma_i^2 : \mathsf{A}^{|\bar{\mathrm{x}}_1^1|} \times \mathsf{A}^{|\bar{\mathrm{x}}_1^2|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{x}}_i^1|} \times \mathsf{A}^{|\bar{\mathrm{x}}_i^2|} \to \mathsf{A}^{|\bar{\mathrm{y}}_i^2|}$.

$\overline{W}$

$\bar{\mathrm{x}}_i^1$, $\bar{\mathrm{x}}_i^2$, $\bar{\mathrm{y}}_i^1$, $\bar{\mathrm{y}}_i^2$, σ_i^1, σ_i^2

Next, we define fingerprint functions that suit the setting of Theorem 7.2.8. In this situation it is convenient to use two complementing kinds of fingerprints. On the one hand, the mappings ν_k, ν'_k assign fingerprints over the set $\mathrm{At}(\overline{W})$ to sequences of tuples $\bar{\mathsf{a}}_1, \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{c}}_k$ with $\bar{\mathsf{a}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}_i^1|}$ and $\bar{\mathsf{c}}_i \in \mathsf{A}^{|\bar{\mathrm{y}}_i^1|}$. The mappings ξ_k, ξ'_k, on the other hand, assign fingerprints over the set $\mathrm{At}(W)$ to sequences of tuples $\bar{\mathsf{b}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{d}}_k$ with $\bar{\mathsf{b}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}_i^2|}$ and $\bar{\mathsf{d}}_i \in \mathsf{A}^{|\bar{\mathrm{y}}_i^2|}$.

Definition 7.2.9 (Fingerprint functions ν_k, ν'_k)**.** *Based on φ and $\mathcal{A}$, we define the family of* fingerprint functions ν_k, ν'_k *with $0 \leq k \leq n$ as follows*

ν_k, ν'_k

$\nu'_n : \mathsf{A}^{|\bar{\mathrm{x}}_1^1|} \times \mathsf{A}^{|\bar{\mathrm{y}}_1^1|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{x}}_n^1|} \times \mathsf{A}^{|\bar{\mathrm{y}}_n^1|} \to \mathcal{P}\mathrm{At}(\overline{W})$ *such that for all tuples $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n$, $\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_n$ and every $A \in \mathrm{At}(\overline{W})$ we have $A(\bar{\mathrm{x}}_1^1, \bar{\mathrm{y}}_1^1, \ldots, \bar{\mathrm{x}}_n^1, \bar{\mathrm{y}}_n^1) \in \nu_n(\bar{\mathsf{a}}_1, \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{a}}_n, \bar{\mathsf{c}}_n)$ if and only if $\mathcal{A} \models A(\bar{\mathsf{a}}_1, \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{a}}_n, \bar{\mathsf{c}}_n)$;*

$\nu_n : \mathsf{A}^{|\bar{\mathrm{x}}_1^1|} \times \mathsf{A}^{|\bar{\mathrm{y}}_1^1|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{x}}_{n-1}^1|} \times \mathsf{A}^{|\bar{\mathrm{y}}_{n-1}^1|} \times \mathsf{A}^{|\bar{\mathrm{x}}_n^1|} \to \mathcal{P}^2\mathrm{At}(\overline{W})$ *such that for all tuples $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n$, $\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{n-1}$ and every $S \in \mathcal{P}\mathrm{At}(\overline{W})$ we have $S \in \nu'_n(\bar{\mathsf{a}}_1, \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{a}}_{n-1}, \bar{\mathsf{c}}_{n-1}, \bar{\mathsf{a}}_n)$ if and only if there is some $\bar{\mathsf{c}}_n$ such that $\nu'_n(\bar{\mathsf{a}}_1, \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{a}}_{n-1}, \bar{\mathsf{c}}_{n-1}, \bar{\mathsf{a}}_n, \bar{\mathsf{c}}_n) = S$;*

$\nu'_{n-1} : \mathsf{A}^{|\bar{\mathrm{x}}_1^1|} \times \mathsf{A}^{|\bar{\mathrm{y}}_1^1|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{x}}_{n-1}^1|} \times \mathsf{A}^{|\bar{\mathrm{y}}_{n-1}^1|} \to \mathcal{P}^3\mathrm{At}(\overline{W})$ *such that for all tuples $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_{n-1}$, $\bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{c}}_{n-1}$ and every $S \in \mathcal{P}^2\mathrm{At}(\overline{W})$ we have $S \in \nu'_{n-1}(\bar{\mathsf{a}}_1, \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{a}}_{n-1}, \bar{\mathsf{c}}_{n-1})$ if and only if there is some $\bar{\mathsf{a}}_n$ such that $\nu_n(\bar{\mathsf{a}}_1, \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{a}}_{n-1}, \bar{\mathsf{c}}_{n-1}, \bar{\mathsf{a}}_n) = S$;*

$\vdots$

$\nu'_1 : \mathsf{A}^{|\bar{\mathrm{x}}_1^1|} \times \mathsf{A}^{|\bar{\mathrm{y}}_1^1|} \to \mathcal{P}^{2n-1}\mathrm{At}(\overline{W})$ *such that for all tuples $\bar{\mathsf{a}}_1, \bar{\mathsf{c}}_1$ and every $S \in \mathcal{P}^{2n-2}\mathrm{At}(\overline{W})$ we have $S \in \nu'_1(\bar{\mathsf{a}}_1, \bar{\mathsf{c}}_1)$ if and only if there exist some $\bar{\mathsf{a}}_2$ such that $\nu_2(\bar{\mathsf{a}}_1, \bar{\mathsf{c}}_1, \bar{\mathsf{a}}_2) = S$.*

$\nu_1 : \mathsf{A}^{|\bar{\mathrm{x}}_1^1|} \to \mathcal{P}^{2n}\mathrm{At}(\overline{W})$ *such that for every tuple $\bar{\mathsf{a}}_1$ and every $S \in \mathcal{P}^{2n-1}\mathrm{At}(\overline{W})$ we have $S \in \nu_1(\bar{\mathsf{a}}_1)$ if and only if there exist some $\bar{\mathsf{c}}_1$ such that $\nu'_1(\bar{\mathsf{a}}_1, \bar{\mathsf{c}}_1) = S$.*

We denote the image *of any fingerprint function ν'_k by*

$$\mathrm{im}(\nu'_k) := \big\{ \nu'_k(\bar{\mathsf{a}}_1, \bar{\mathsf{c}}_1, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{c}}_k) \;\big|\; \bar{\mathsf{a}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}_i^1|},\ \bar{\mathsf{c}}_i \in \mathsf{A}^{|\bar{\mathrm{y}}_i^1|} \text{ for every } i \big\}.$$

$\mathrm{im}(\nu'_k)$,

Analogously, we denote the image *of any fingerprint function ν_k by $\mathrm{im}(\nu_k)$. Given a strategy σ, we denote the* image under σ *of any fingerprint function ν'_k by*

$\mathrm{im}(\nu_k)$

$$\mathrm{im}_\sigma(\nu'_k) := \big\{ \nu'_k\big(\bar{\mathsf{a}}_1, \sigma_1^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{a}}_k, \sigma_k^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{b}}_k)\big) \;\big|\; \bar{\mathsf{a}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}_i^1|},\ \bar{\mathsf{b}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}_i^2|} \text{ for every } i \big\}.$$

$\mathrm{im}_\sigma(\nu'_k)$,

Analogously, we denote the image under σ *of any fingerprint function ν_k by* $\mathrm{im}_\sigma(\nu_k)$.

$\mathrm{im}_\sigma(\nu_k)$

Definition 7.2.10 (Fingerprint functions ξ_k, ξ'_k)**.** *Based on φ and $\mathcal{A}$, we define the family of* fingerprint functions ξ_k, ξ'_k *with $0 \leq k \leq n$ as follows*

ξ_k, ξ'_k

$\xi'_n : \mathsf{A}^{|\bar{\mathrm{x}}_1^2|} \times \mathsf{A}^{|\bar{\mathrm{y}}_1^2|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{x}}_n^2|} \times \mathsf{A}^{|\bar{\mathrm{y}}_n^2|} \to \mathcal{P}\mathrm{At}(W)$ *such that for all tuples $\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_n$, $\bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{d}}_n$ and every $A \in \mathrm{At}(W)$ we have $A(\bar{\mathrm{x}}_1^2, \bar{\mathrm{y}}_1^2, \ldots, \bar{\mathrm{x}}_n^2, \bar{\mathrm{y}}_n^2) \in \xi_n(\bar{\mathsf{b}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{b}}_n, \bar{\mathsf{d}}_n)$ if and only if $\mathcal{A} \models A(\bar{\mathsf{b}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{b}}_n, \bar{\mathsf{d}}_n)$;*

$\xi_n : \mathsf{A}^{|\bar{\mathrm{x}}_1^2|} \times \mathsf{A}^{|\bar{\mathrm{y}}_1^2|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{x}}_{n-1}^2|} \times \mathsf{A}^{|\bar{\mathrm{y}}_{n-1}^2|} \times \mathsf{A}^{|\bar{\mathrm{x}}_n^2|} \to \mathcal{P}^2\mathrm{At}(W)$ *such that for all tuples $\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_n$, $\bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{d}}_{n-1}$ and every $S \in \mathcal{P}\mathrm{At}(W)$ we have $S \in \xi_n(\bar{\mathsf{b}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{b}}_{n-1}, \bar{\mathsf{d}}_{n-1}, \bar{\mathsf{b}}_n)$ if and only if there is some $\bar{\mathsf{d}}_n$ such that $\xi'_n(\bar{\mathsf{b}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{b}}_{n-1}, \bar{\mathsf{d}}_{n-1}, \bar{\mathsf{b}}_n, \bar{\mathsf{d}}_n) = S$;*

$\xi'_{n-1} : \mathsf{A}^{|\bar{\mathsf{x}}^2_1|} \times \mathsf{A}^{|\bar{\mathsf{y}}^2_1|} \times \ldots \times \mathsf{A}^{|\bar{\mathsf{x}}^2_{n-1}|} \times \mathsf{A}^{|\bar{\mathsf{y}}^2_{n-1}|} \to \mathcal{P}^3\mathrm{At}(W)$ *such that for all tuples* $\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_{n-1}$, $\bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{d}}_{n-1}$ *and every* $S \in \mathcal{P}^2\mathrm{At}(W)$ *we have* $S \in \xi'_{n-1}(\bar{\mathsf{b}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{b}}_{n-1}, \bar{\mathsf{d}}_{n-1})$ *if and only if there is some* $\bar{\mathsf{b}}_n$ *such that* $\xi_n(\bar{\mathsf{b}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{b}}_{n-1}, \bar{\mathsf{d}}_{n-1}, \bar{\mathsf{b}}_n) = S$*;*

$\vdots$

$\xi'_1 : \mathsf{A}^{|\bar{\mathsf{x}}^2_1|} \times \mathsf{A}^{|\bar{\mathsf{y}}^2_1|} \to \mathcal{P}^{2n-1}\mathrm{At}(W)$ *such that for all tuples* $\bar{\mathsf{b}}_1, \bar{\mathsf{d}}_1$ *and every* $S \in \mathcal{P}^{2n-2}\mathrm{At}(W)$ *we have* $S \in \xi'_1(\bar{\mathsf{b}}_1, \bar{\mathsf{d}}_1)$ *if and only if there exist some* $\bar{\mathsf{b}}_2$ *such that* $\xi_2(\bar{\mathsf{b}}_1, \bar{\mathsf{d}}_1, \bar{\mathsf{b}}_2) = S$*.*

$\xi_1 : \mathsf{A}^{|\bar{\mathsf{x}}^2_1|} \to \mathcal{P}^{2n}\mathrm{At}(W)$ *such that for every tuple* $\bar{\mathsf{b}}_1$ *and every* $S \in \mathcal{P}^{2n-1}\mathrm{At}(W)$ *we have* $S \in \xi_1(\bar{\mathsf{b}}_1)$ *if and only if there exist some* $\bar{\mathsf{d}}_1$ *such that* $\xi'_1(\bar{\mathsf{b}}_1, \bar{\mathsf{d}}_1) = S$*.*

$\mathrm{im}(\xi'_k)$, $\mathrm{im}(\xi_k)$ *In analogy to Definition 7.2.9, we denote the* image *of any fingerprint function* ξ_k *by* $\mathrm{im}(\xi'_k)$ *and the* image *of any* ξ_k *by* $\mathrm{im}(\xi_k)$*. Moreover, given any strategy* σ*, we denote the* image under σ *of any fingerprint function* ξ_k *by*

$\mathrm{im}_\sigma(\xi'_k)$, $\mathrm{im}_\sigma(\xi_k)$ $\mathrm{im}_\sigma(\xi'_k) := \big\{\xi'_k\big(\bar{\mathsf{b}}_1, \sigma^2_1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{b}}_k, \sigma^2_k(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{b}}_k)\big) \;\big|\; \bar{\mathsf{a}}_i \in \mathsf{A}^{|\bar{\mathsf{x}}^1_i|},\ \bar{\mathsf{b}}_i \in \mathsf{A}^{|\bar{\mathsf{x}}^2_i|}$ *for every* $i\big\}$*, and the* image under σ *of any fingerprint function* ξ_k *is denoted by* $\mathrm{im}_\sigma(\xi_k)$*.*

Next we show that, staring from any given strategy, we can construct a strategy with a similar outcome that is uniform with respect to the just defined fingerprint functions.

Lemma 7.2.11. *For every strategy* $\sigma = \langle \sigma_1, \ldots, \sigma_n \rangle$ *there is a strategy* $\tau = \langle \tau_1, \ldots, \tau_n \rangle$ *that satisfies the following conditions.*

ν-ξ-uniformity *(a)* τ *is required to be* ν-ξ-uniform*. That is, for every* k *and all tuples* $\bar{\mathsf{a}}_1, \bar{\mathsf{a}}'_1 \in \mathsf{A}^{|\bar{\mathsf{x}}^1_1|}, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{a}}'_k \in \mathsf{A}^{|\bar{\mathsf{x}}^1_k|} \in \mathsf{A}^{|\bar{\mathsf{x}}^1_k|}$, $\bar{\mathsf{b}}_1, \bar{\mathsf{b}}'_1 \in \mathsf{A}^{|\bar{\mathsf{x}}^2_1|}, \ldots, \bar{\mathsf{b}}_k, \bar{\mathsf{b}}'_k \in \mathsf{A}^{|\bar{\mathsf{x}}^2_k|}$ *we observe*

$$\tau^1_k(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{b}}_k) = \tau^1_k(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}'_1, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{b}}'_k)$$

and

$$\tau^2_k(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{b}}_k) = \tau^2_k(\bar{\mathsf{a}}'_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}'_k, \bar{\mathsf{b}}_k)$$

whenever we have

$$\begin{aligned}&\nu_\ell\big(\bar{\mathsf{a}}_1, \tau^1_1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{a}}_{\ell-1}, \tau^1_{\ell-1}(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{\ell-1}, \bar{\mathsf{b}}_{\ell-1}), \bar{\mathsf{a}}_\ell\big)\\ &= \nu_\ell\big(\bar{\mathsf{a}}'_1, \tau^1_1(\bar{\mathsf{a}}'_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{a}}'_{\ell-1}, \tau^1_{\ell-1}(\bar{\mathsf{a}}'_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}'_{\ell-1}, \bar{\mathsf{b}}_{\ell-1}), \bar{\mathsf{a}}'_\ell\big)\end{aligned}$$

and

$$\begin{aligned}&\xi_\ell\big(\bar{\mathsf{b}}_1, \tau^1_1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{b}}_{\ell-1}, \tau^1_{\ell-1}(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{\ell-1}, \bar{\mathsf{b}}_{\ell-1}), \bar{\mathsf{b}}_\ell\big)\\ &= \xi_\ell\big(\bar{\mathsf{b}}'_1, \tau^1_1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}'_1), \ldots, \bar{\mathsf{b}}'_{\ell-1}, \tau^1_{\ell-1}(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}'_1, \ldots, \bar{\mathsf{a}}_{\ell-1}, \bar{\mathsf{b}}'_{\ell-1}), \bar{\mathsf{b}}'_\ell\big)\end{aligned}$$

for every ℓ*,* $1 \leq \ell \leq k$

(b) $\mathrm{Out}_\tau \subseteq \mathrm{Out}_\sigma$*.*

$\underline{\mathsf{A}}_i$, $\alpha_{i,\langle\ldots\rangle}$, $\overline{S}^{(k)}$, $\overline{T}^{(k)}$ *Proof.* For $i = 1, \ldots, n$ we define $\underline{\mathsf{A}}_i$ as abbreviation of $\mathsf{A}^{|\bar{\mathsf{x}}^1_1|} \times \mathsf{A}^{|\bar{\mathsf{x}}^2_1|} \times \ldots \times \mathsf{A}^{|\bar{\mathsf{x}}^1_i|} \times \mathsf{A}^{|\bar{\mathsf{x}}^2_i|}$. We construct certain representatives $\alpha_{k,\langle\overline{S}^{(k)},\overline{T}^{(k)}\rangle} \in \underline{\mathsf{A}}_k$ inductively as follows. The $\overline{S}^{(k)}$ and $\overline{T}^{(k)}$ stand for a sequences $S_1 \ldots S_{2k-1}$ and $T_1 \ldots T_{2k-1}$ of $2k-1$ fingerprints each, satisfying $S_{2k-1} \in S_{2k-2} \in \ldots \in S_1$ and $T_{2k-1} \in T_{2k-2} \in \ldots \in T_1$.

$\underline{\mathsf{A}}_{k,\langle\overline{S}^{(k)},\overline{T}^{(k)}\rangle}$ Let $k = 1$. We partition $\underline{\mathsf{A}}_1$ into sets $\underline{\mathsf{A}}_{1,\langle S_1,T_1\rangle}$ with $S_1 \in \mathrm{im}(\nu_1)$ and $T_1 \in \mathrm{im}(\xi_1)$ by setting $\underline{\mathsf{A}}_{1,\langle S_1,T_1\rangle} := \big\{\langle \bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1\rangle \in \mathsf{A}^{|\bar{\mathsf{x}}^1_1|} \times \mathsf{A}^{|\bar{\mathsf{x}}^2_1|} \;\big|\; \nu_1\big(\bar{\mathsf{a}}_1\big) = S_1 \text{ and } \xi_1\big(\bar{\mathsf{b}}_1\big) = T_1\big\}$. We pick one representative $\alpha_{1,\langle S_1,T_1\rangle} \in \underline{\mathsf{A}}_{1,\langle S_1,T_1\rangle}$ from every set $\underline{\mathsf{A}}_{1,\langle S_1,T_1\rangle}$.

Let $k > 1$. We construct subsets $\underline{\mathsf{A}}_{k,\langle\overline{S}^{(k)},\overline{T}^{(k)}\rangle} \subseteq \underline{\mathsf{A}}_k$ with $S_{2i} \in S_{2i-1} \in \mathrm{im}(\nu_i)$ and $T_{2i} \in T_{2i-1} \in$

$\mathrm{im}(\xi_i)$ for every i, $1 \leq i \leq k$, by setting $\underline{\mathsf{A}}_{k,\langle \overline{S}^{(k)}, \overline{T}^{(k)} \rangle} :=$

$$\Big\{ \big\langle \bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}_{k-1}, \bar{\mathsf{a}}_k, \bar{\mathsf{b}}_k \big\rangle \;\Big|$$
$$\bar{\mathsf{a}}_k \in \mathsf{A}^{|\bar{\mathrm{x}}^1_k|},\ \bar{\mathsf{b}}_k \in \mathsf{A}^{|\bar{\mathrm{x}}^2_k|} \text{ and there is some}$$
$$\alpha_{k-1,\langle \overline{S}^{(k-1)}, \overline{T}^{(k-1)} \rangle} = \big\langle \bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}_{k-1} \big\rangle,$$
$$\bar{\mathsf{c}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}^1_i|} \text{ and } \bar{\mathsf{d}}_i \in \mathsf{A}^{|\bar{\mathrm{x}}^2_i|}, \text{ for every } i, \text{ such that}$$
$$\nu'_{k-1}\big(\bar{\mathsf{c}}_1, \sigma^1_1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{c}}_{k-1}, \sigma^1_{k-1}(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}_{k-1})\big) = S_{2k-2},$$
$$\xi'_{k-1}\big(\bar{\mathsf{d}}_1, \sigma^2_1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{d}}_{k-1}, \sigma^2_{k-1}(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}_{k-1})\big) = T_{2k-2},$$
$$\nu_k\big(\bar{\mathsf{c}}_1, \sigma^1_1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{c}}_{k-1}, \sigma^1_{k-1}(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}_{k-1}), \bar{\mathsf{a}}_k\big) = S_{2k-1}, \text{ and}$$
$$\xi_k\big(\bar{\mathsf{d}}_1, \sigma^2_1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{d}}_{k-1}, \sigma^2_{k-1}(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}_{k-1}), \bar{\mathsf{b}}_k\big) = T_{2k-1} \Big\}\ .$$

We pick one representative $\alpha_{k,\langle \overline{S}^{(k)}, \overline{T}^{(k)} \rangle}$ from each nonempty $\underline{\mathsf{A}}_{k,\langle \overline{S}^{(k)}, \overline{T}^{(k)} \rangle}$.

Having all the representatives $\alpha_{k,\langle \overline{S}^{(k)}, \overline{T}^{(k)} \rangle}$ at hand, we inductively construct τ, starting from τ^1_1, τ^2_1 and going to τ^1_n, τ^2_n.

Let $k = 1$. Consider any representative $\langle \bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1 \rangle := \alpha_{1,\langle S_1, T_1 \rangle}$ for any fingerprints S_1, T_1. Let $S_2 := \nu'_1\big(\bar{\mathsf{c}}_1, \sigma^1_1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1)\big)$ and $T_2 := \xi'_1\big(\bar{\mathsf{d}}_1, \sigma^2_1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1)\big)$.

For any tuple $\bar{\mathsf{a}}_1 \in \mathsf{A}^{|\bar{\mathrm{x}}^1_1|}$ with $\nu_1\big(\bar{\mathsf{a}}_1\big) = S_1$ we define $\tau^1_1(\bar{\mathsf{a}}_1, \bar{\mathsf{d}}_1)$ as follows. Since $S_2 \in S_1 = \nu_1\big(\bar{\mathsf{a}}_1\big)$, there is some tuple $\bar{\mathsf{e}}_1$ for which $\nu'_1\big(\bar{\mathsf{a}}_1, \bar{\mathsf{e}}_1\big) = S_2$. We set $\tau^1_1(\bar{\mathsf{a}}_1, \bar{\mathsf{d}}_1) := \bar{\mathsf{e}}_1$. In case of $\bar{\mathsf{a}}_1 = \bar{\mathsf{c}}_1$, we make sure that $\bar{\mathsf{e}}_1 = \sigma^1_1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1)$, i.e. we set $\tau^1_1(\alpha_{1,\langle S_1,T_1 \rangle}) := \sigma^1_1(\alpha_{1,\langle S_1,T_1 \rangle})$. Hence, we get $\nu'_1\big(\bar{\mathsf{a}}_1, \tau^1_1(\bar{\mathsf{a}}_1, \bar{\mathsf{d}}_1)\big) = S_2$.

Symmetriaclly, for any tuple $\bar{\mathsf{b}}_1 \in \mathsf{A}^{|\bar{\mathrm{x}}^2_1|}$ with $\xi_1\big(\bar{\mathsf{b}}_1\big) = T_1$ we define $\tau^2_1(\bar{\mathsf{c}}_1, \bar{\mathsf{b}}_1)$ as follows. Since $T_2 \in T_1 = \xi_1\big(\bar{\mathsf{b}}_1\big)$, there is some tuple $\bar{\mathsf{f}}_1$ for which $\xi'_1\big(\bar{\mathsf{b}}_1, \bar{\mathsf{f}}_1\big) = T_2$. We set $\tau^2_1(\bar{\mathsf{c}}_1, \bar{\mathsf{b}}_1) := \bar{\mathsf{f}}_1$. In case of $\bar{\mathsf{b}}_1 = \bar{\mathsf{d}}_1$, we make sure that $\bar{\mathsf{f}}_1 = \sigma^2_1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1)$, i.e. we set $\tau^2_1(\alpha_{1,\langle S_1,T_1 \rangle}) := \sigma^2_1(\alpha_{1,\langle S_1,T_1 \rangle})$. Hence, we get $\xi'_1\big(\bar{\mathsf{b}}_1, \tau^2_1(\bar{\mathsf{c}}_1, \bar{\mathsf{b}}_1)\big) = T_2$.

Now, consider any two tuples $\bar{\mathsf{a}}_1 \in \mathsf{A}^{|\bar{\mathrm{x}}^1_1|}$, $\bar{\mathsf{b}}_1 \in \mathsf{A}^{|\bar{\mathrm{x}}^2_1|}$, and let $S_1 := \nu_1(\bar{\mathsf{a}}_1)$ and $T_1 := \xi_1(\bar{\mathsf{b}}_1)$. Moreover, let $\langle \bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1 \rangle := \alpha_{1,\langle S_1,T_1 \rangle}$. We set and $\tau^1_1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1) := \tau^1_1(\bar{\mathsf{a}}_1, \bar{\mathsf{d}}_1)$ and $\tau^2_1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1) := \tau^2_1(\bar{\mathsf{c}}_1, \bar{\mathsf{b}}_1)$.

Let $k > 1$. Consider any representative $\langle \bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_k, \bar{\mathsf{d}}_k \rangle := \alpha_{1,\langle \overline{S}^{(k)}, \overline{T}^{(k)} \rangle}$ for any two sequences of fingerprints $\overline{S}^{(k)}, \overline{T}^{(k)}$. Let $S_{2k} := \nu'_k\big(\bar{\mathsf{c}}_1, \sigma^1_1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{c}}_k, \sigma^1_k(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_k, \bar{\mathsf{d}}_k)\big)$ and $T_{2k} := \xi'_k\big(\bar{\mathsf{d}}_1, \sigma^2_1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{d}}_k, \sigma^2_k(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_k, \bar{\mathsf{d}}_k)\big)$.

For any sequence of tuples $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k$ with
$$\nu_\ell\big(\bar{\mathsf{a}}_1, \tau^1_1(\bar{\mathsf{a}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{a}}_{\ell-1}, \tau^1_{\ell-1}(\bar{\mathsf{a}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{a}}_{\ell-1}, \bar{\mathsf{d}}_{\ell-1}), \bar{\mathsf{a}}_\ell\big) = S_{2\ell-1}$$
for every ℓ, $1 \leq \ell \leq k$, we define $\tau^1_1(\bar{\mathsf{a}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{d}}_k)$ as follows. Since $S_{2k} \in S_{2k-1}$, there is some tuple $\bar{\mathsf{e}}_k$ for which
$$\nu'_k\big(\bar{\mathsf{a}}_1, \tau^1_1(\bar{\mathsf{a}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{a}}_{k-1}, \tau^1_{k-1}(\bar{\mathsf{a}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{a}}_{k-1}, \bar{\mathsf{d}}_{k-1}), \bar{\mathsf{a}}_k, \bar{\mathsf{e}}_k\big) = S_{2k}.$$
We set $\tau^1_k(\bar{\mathsf{a}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{d}}_k) := \bar{\mathsf{e}}_k$. In case of $\bar{\mathsf{a}}_i = \bar{\mathsf{c}}_i$ for every i, we make sure that $\bar{\mathsf{e}}_k = \sigma^1_k(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_k, \bar{\mathsf{d}}_k)$, i.e. we set $\tau^1_k(\alpha_{k,\langle \overline{S}^{(k)}, \overline{T}^{(k)} \rangle}) := \sigma^1_k(\alpha_{k,\langle \overline{S}^{(k)}, \overline{T}^{(k)} \rangle})$. Hence, we get
$$\nu'_k\big(\bar{\mathsf{a}}_1, \tau^1_1(\bar{\mathsf{a}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{a}}_{k-1}, \tau^1_{k-1}(\bar{\mathsf{a}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{a}}_{k-1}, \bar{\mathsf{d}}_{k-1}), \bar{\mathsf{a}}_k, \tau^1_k(\bar{\mathsf{a}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{d}}_k)\big) = S_{2k}.$$
Symmetrically, for any sequence of tuples $\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_k$ with
$$\xi_\ell\big(\bar{\mathsf{b}}_1, \tau^2_1(\bar{\mathsf{c}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{b}}_{\ell-1}, \tau^2_{\ell-1}(\bar{\mathsf{c}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{c}}_{\ell-1}, \bar{\mathsf{b}}_{\ell-1}), \bar{\mathsf{b}}_\ell\big) = T_{2\ell-1}$$
for every ℓ, $1 \leq \ell \leq k$, we define $\tau^2_k(\bar{\mathsf{c}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{c}}_k, \bar{\mathsf{b}}_k)$ as follows. Since $T_{2k} \in T_{2k-1}$, there is some tuple $\bar{\mathsf{f}}_k$ for which
$$\xi'_k\big(\bar{\mathsf{b}}_1, \tau^2_1(\bar{\mathsf{c}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{a}}_{k-1}, \tau^2_{k-1}(\bar{\mathsf{c}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{b}}_{k-1}), \bar{\mathsf{b}}_k, \bar{\mathsf{f}}_k\big) = T_{2k}.$$
We set $\tau^2_k(\bar{\mathsf{c}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{c}}_k, \bar{\mathsf{b}}_k) := \bar{\mathsf{f}}_k$. In case of $\bar{\mathsf{b}}_i = \bar{\mathsf{d}}_i$ for every i, we make sure that

$\bar{\mathsf{f}}_k = \sigma_k^2(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_k, \bar{\mathsf{d}}_k)$, i.e. we set $\tau_k^2(\alpha_{k,\langle \overline{S}^{(k)}, \overline{T}^{(k)}\rangle}) := \sigma_k^2(\alpha_{k,\langle \overline{S}^{(k)}, \overline{T}^{(k)}\rangle})$. Hence, we get

$$\xi_k'\big(\bar{\mathsf{b}}_1, \tau_1^2(\bar{\mathsf{c}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{b}}_{k-1}, \tau_{k-1}^2(\bar{\mathsf{c}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{b}}_{k-1}), \bar{\mathsf{b}}_k, \tau_{k-1}^2(\bar{\mathsf{c}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{c}}_k, \bar{\mathsf{b}}_k)\big) = T_{2k}.$$

Now, consider any sequence of tuples $\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{b}}_k$ with $\bar{\mathsf{a}}_i \in \mathsf{A}^{|\bar{\mathsf{x}}_i^1|}$ and $\bar{\mathsf{b}}_i \in \mathsf{A}^{|\bar{\mathsf{x}}_i^2|}$ for every i. Let $\overline{S}^{(k)} := S_1 \ldots S_{2k-1}$ be a sequence of fingerprints such that

$$S_{2i-1} := \nu_i\big(\bar{\mathsf{a}}_1, \tau_1^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{a}}_{i-1}, \tau_{i-1}^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{i-1}, \bar{\mathsf{b}}_{i-1}), \bar{\mathsf{a}}_i\big)$$

and

$$S_{2i} := \nu_i'\big(\bar{\mathsf{a}}_1, \tau_1^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{a}}_{i-1}, \tau_{i-1}^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{i-1}, \bar{\mathsf{b}}_{i-1}), \bar{\mathsf{a}}_i, \tau_i^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_i, \bar{\mathsf{b}}_i)\big)$$

for every i, $1 \leq i \leq k-1$, and, moreover,

$$S_{2k-1} := \nu_k\big(\bar{\mathsf{a}}_1, \tau_1^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{a}}_{k-1}, \tau_{k-1}^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{k-1}, \bar{\mathsf{b}}_{k-1}), \bar{\mathsf{a}}_k\big).$$

In addition, let $\overline{T}^{(k)} := T_1 \ldots T_{2k-1}$ be a sequence of fingerprints such that

$$T_{2i-1} := \xi_i\big(\bar{\mathsf{b}}_1, \tau_1^2(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{b}}_{i-1}, \tau_{i-1}^2(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{i-1}, \bar{\mathsf{b}}_{i-1}), \bar{\mathsf{b}}_i\big)$$

and

$$T_{2i} := \xi_i'\big(\bar{\mathsf{b}}_1, \tau_1^2(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{b}}_{i-1}, \tau_{i-1}^2(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{i-1}, \bar{\mathsf{b}}_{i-1}), \bar{\mathsf{b}}_i, \tau_i^2(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_i, \bar{\mathsf{b}}_i)\big)$$

for every i, $1 \leq i \leq k-1$, and, moreover,

$$T_{2k-1} := \xi_k\big(\bar{\mathsf{b}}_1, \tau_1^2(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{b}}_{k-1}, \tau_{k-1}^2(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{k-1}, \bar{\mathsf{b}}_{k-1}), \bar{\mathsf{b}}_k\big).$$

Suppose there exists a representative $\langle \bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_k, \bar{\mathsf{d}}_k\rangle := \alpha_{k,\langle \overline{S}^{(k)}, \overline{T}^{(k)}\rangle}$ — we show in Claim II that this is always the case. Then, we set $\tau_k^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{b}}_k) := \tau_k^1(\bar{\mathsf{a}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{d}}_k)$ and $\tau_k^2(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{b}}_k) := \tau_k^2(\bar{\mathsf{c}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{c}}_k, \bar{\mathsf{b}}_k)$.

Claim Ia: For all k, $1 \leq k \leq n$, we have $\mathrm{im}_\tau\big(\nu_k\big) \subseteq \mathrm{im}_\sigma\big(\nu_k\big)$ (where we consider $\mathrm{im}_\tau(\nu_k)$ to be defined such that any value $\nu_k(\ldots)$ can only enter $\mathrm{im}_\tau(\nu_k)$ if all of its arguments are defined).

Proof: Fix some ν_k and let $S \in \mathrm{im}_\tau(\nu_k)$. Then, there are tuples $\bar{\mathsf{a}}_1 \in \mathsf{A}^{|\bar{\mathsf{x}}_1^1|}, \ldots, \bar{\mathsf{a}}_k \in \mathsf{A}^{|\bar{\mathsf{x}}_k^1|}$, $\bar{\mathsf{b}}_1 \in \mathsf{A}^{|\bar{\mathsf{x}}_1^2|}, \ldots, \bar{\mathsf{b}}_{k-1} \in \mathsf{A}^{|\bar{\mathsf{x}}_{k-1}^2|}$ such that $\tau_1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \tau_{k-1}(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{k-1}, \bar{\mathsf{b}}_{k-1})$ are defined and we have

$$S = \nu_k\big(\bar{\mathsf{a}}_1, \tau_1^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{a}}_{k-1}, \tau_{k-1}^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{k-1}, \bar{\mathsf{b}}_{k-1}), \bar{\mathsf{a}}_k\big).$$

By construction of τ, the mapping τ_{k-1}^1 is defined in such a way that

$$\begin{aligned}
&\nu_{k-1}'\big(\bar{\mathsf{a}}_1, \tau_1^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{a}}_{k-1}, \tau_{k-1}^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{k-1}, \bar{\mathsf{b}}_{k-1})\big)\\
&= \nu_{k-1}'\big(\bar{\mathsf{c}}_1, \tau_1^1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{c}}_{k-1}, \tau_{k-1}^1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}_{k-1})\big)\\
&= \nu_{k-1}'\big(\bar{\mathsf{c}}_1, \sigma_1^1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{c}}_{k-1}, \sigma_{k-1}^1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}_{k-1})\big)
\end{aligned}$$

for a certain representative $\langle \bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}_{k-1}\rangle = \alpha_{k-1,\langle \overline{S}^{(k-1)}, \overline{T}^{(k-1)}\rangle}$.

Since $S \in \nu_{k-1}'\big(\bar{\mathsf{a}}_1, \tau_1^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{a}}_{k-1}, \tau_{k-1}^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{k-1}, \bar{\mathsf{b}}_{k-1})\big)$, the definition of ν_{k-1}' entails that there is some tuple $\bar{\mathsf{a}}_k' \in \mathsf{A}^{|\bar{\mathsf{x}}_k^1|}$ such that

$$\nu_k\big(\bar{\mathsf{c}}_1, \sigma_1^1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{c}}_{k-1}, \sigma_{k-1}^1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}_{k-1}), \bar{\mathsf{a}}_k'\big) = S.$$

Consequently, we have $S \in \mathrm{im}_\sigma(\nu_k)$. ◊

Claim Ib: For all k, $1 \leq k \leq n$, we have $\mathrm{im}_\tau\big(\xi_k\big) \subseteq \mathrm{im}_\sigma\big(\xi_k\big)$ (where we consider $\mathrm{im}_\tau(\xi_k)$ to be defined such that any value $\xi_k(\ldots)$ can only enter $\mathrm{im}_\tau(\xi_k)$ if all of its arguments are defined).

Proof: The proof is similar to the proof of Claim Ia. ◊

Claim II: For every k, $1 \leq k \leq n$, and all tuples $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k$, $\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_k$ there is a (unique) representative $\alpha_{k,\langle \overline{S}^{(k)}, \overline{T}^{(k)}\rangle}$ such that

$$\begin{aligned}
S_1 &= \nu_1\big(\bar{\mathsf{a}}_1\big)\ ,\\
S_2 &= \nu_1'\big(\bar{\mathsf{a}}_1, \tau_1^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1)\big)\ ,\\
&\vdots\\
S_{2k-1} &= \nu_k\big(\bar{\mathsf{a}}_1, \tau_1^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{a}}_{k-1}, \tau_{k-1}^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{k-1}, \bar{\mathsf{b}}_{k-1}), \bar{\mathsf{a}}_k\big)\ ,
\end{aligned}$$

and

$$\begin{aligned}
T_1 &= \xi_1\big(\bar{\mathsf{b}}_1\big) \ , \\
T_2 &= \xi'_1\big(\bar{\mathsf{b}}_1, \tau_1^2(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1)\big) \ , \\
&\vdots \\
T_{2k-1} &= \xi_k\big(\bar{\mathsf{b}}_1, \tau_1^2(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{b}}_{k-1}, \tau_{k-1}^2(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{k-1}, \bar{\mathsf{b}}_{k-1}), \bar{\mathsf{b}}_k\big) \ .
\end{aligned}$$

Proof: We proceed by induction on k.

Let $k = 1$. Consider any pair of tuples $\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1$ and set $S_1 := \nu_1\big(\bar{\mathsf{a}}_1\big)$ and $T_1 := \xi_1\big(\bar{\mathsf{b}}_1\big)$. Obviously, we have $S_1 \in \mathrm{im}(\nu_1)$ and $T_1 \in \mathrm{im}(\xi_1)$ and $\langle \bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1 \rangle \in \underline{\mathsf{A}}_{1,\langle S_1, T_1\rangle}$. Since the set is nonempty, there is a representative $\alpha_{1,\langle S_1,T_1\rangle} \in \underline{\mathsf{A}}_{1,\langle S_1,T_1\rangle}$.

Let $k > 1$. Consider any sequence of tuples $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k$, $\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_k$. Let the sequences $\overline{S}^{(k)} := S_1 \ldots S_{2k-1}$ and $\overline{T}^{(k)} := T_1 \ldots T_{2k-1}$ be defined as indicated in the claim. By Claim Ia and Claim Ib, we have $S_{2j-1} \in \mathrm{im}_\tau(\nu_j) \subseteq \mathrm{im}_\sigma(\nu_j) \subseteq \mathrm{im}(\nu_j)$ and $T_{2j-1} \in \mathrm{im}_\tau(\xi_j) \subseteq \mathrm{im}_\sigma(\xi_j) \subseteq \mathrm{im}(\xi_j)$ for every j, $1 \leq j \leq k$. Moreover, the definitions of the ν_j and the ξ_j entail that $S_{2j} \in S_{2j-1}$ and $T_{2j} \in T_{2j-1}$ for every j, $1 \leq j \leq k$. Therefore, we have constructed the subset $\underline{\mathsf{A}}_{k,\langle \overline{S}^{(k)}, \overline{T}^{(k)}\rangle} \subseteq \underline{\mathsf{A}}_k$ when defining representatives. We next show that this set is not empty.

By induction, there is some representative $\langle \bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}_{k-1}\rangle := \alpha_{k-1,\langle \overline{S}^{(k-1)}, \overline{T}^{(k-1)}\rangle}$. More precisely, the definition of τ entails

$$\begin{aligned}
S_{2k-3} &= \nu_{k-1}\big(\bar{\mathsf{c}}_1, \sigma_1^1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{c}}_{k-2}, \sigma_{k-2}^1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-2}, \bar{\mathsf{d}}_{k-2}), \bar{\mathsf{c}}_{k-1}\big) \\
&= \nu_{k-1}\big(\bar{\mathsf{c}}_1, \tau_1^1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{c}}_{k-2}, \tau_{k-2}^1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-2}, \bar{\mathsf{d}}_{k-2}), \bar{\mathsf{c}}_{k-1}\big) \ ,
\end{aligned}$$

and

$$\begin{aligned}
T_{2k-3} &= \xi_{k-1}\big(\bar{\mathsf{d}}_1, \sigma_1^2(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{d}}_{k-2}, \sigma_{k-2}^2(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-2}, \bar{\mathsf{d}}_{k-2}), \bar{\mathsf{d}}_{k-1}\big) \\
&= \xi_{k-1}\big(\bar{\mathsf{d}}_1, \tau_1^1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{d}}_{k-2}, \tau_{k-2}^2(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-2}, \bar{\mathsf{d}}_{k-2}), \bar{\mathsf{d}}_{k-1}\big) \ .
\end{aligned}$$

The mappings τ_{k-1}^1 and τ_{k-1}^2 are defined in such a way that

$$\begin{aligned}
S_{2k-2} &= \nu'_{k-1}\big(\bar{\mathsf{a}}_1, \tau_1^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{a}}_{k-1}, \tau_{k-1}^1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{k-1}, \bar{\mathsf{b}}_{k-1})\big) \\
&= \nu'_{k-1}\big(\bar{\mathsf{a}}_1, \tau_1^1(\bar{\mathsf{a}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{a}}_{k-1}, \tau_{k-1}^1(\bar{\mathsf{a}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{a}}_{k-1}, \bar{\mathsf{d}}_{k-1})\big) \\
&= \nu'_{k-1}\big(\bar{\mathsf{c}}_1, \sigma_1^1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{c}}_{k-1}, \sigma_{k-1}^1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}_{k-1})\big)
\end{aligned}$$

and

$$\begin{aligned}
T_{2k-2} &= \xi'_{k-1}\big(\bar{\mathsf{b}}_1, \tau_1^2(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{b}}_{k-1}, \tau_{k-1}^2(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{k-1}, \bar{\mathsf{b}}_{k-1})\big) \\
&= \xi'_{k-1}\big(\bar{\mathsf{b}}_1, \tau_1^2(\bar{\mathsf{c}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{b}}_{k-1}, \tau_{k-1}^2(\bar{\mathsf{c}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{b}}_{k-1})\big) \\
&= \xi'_{k-1}\big(\bar{\mathsf{d}}_1, \sigma_1^2(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{d}}_{k-1}, \sigma_{k-1}^2(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}_{k-1})\big)
\end{aligned}$$

By definition of ν'_{k-1} and ξ'_{k-1} and the fact that $S_{2k-1} \in S_{2k-2}$ and $T_{2k-1} \in T_{2k-2}$, there are tuples $\bar{\mathsf{a}}'_k, \bar{\mathsf{b}}'_k$ such that

$$S_{2k-1} = \nu_k\big(\bar{\mathsf{c}}_1, \sigma_1^1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{c}}_{k-1}, \sigma_{k-1}^1(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}_{k-1}), \bar{\mathsf{a}}'_k\big)$$

and

$$T_{2k-1} = \xi_k\big(\bar{\mathsf{d}}_1, \sigma_1^2(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{d}}_{k-1}, \sigma_{k-1}^2(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}_{k-1}), \bar{\mathsf{b}}'_k\big) \ .$$

Therefore, we have $\langle \bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{k-1}, \bar{\mathsf{d}}_{k-1}, \bar{\mathsf{a}}'_k, \bar{\mathsf{b}}'_k\rangle \in \underline{\mathsf{A}}_{k,\langle \overline{S}^{(k)}, \overline{T}^{(k)}\rangle}$. Hence, $\underline{\mathsf{A}}_{k,\langle \overline{S}^{(k)}, \overline{T}^{(k)}\rangle}$ is not empty. This means we have defined some — indeed, exactly one — representative $\alpha_{k,\langle \overline{S}^{(k)}, \overline{T}^{(k)}\rangle}$. ◊

Claim III: τ satisfies (a), i.e. it is ν-ξ-uniform.

Proof: By construction of τ. ◊

It remains to show $\mathrm{Out}_\tau \subseteq \mathrm{Out}_\sigma$. Let $U \in \mathrm{Out}_\tau$, i.e. there exist tuples $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_n$, $\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_n$ such that $U = \mathrm{out}_\tau(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_n, \bar{\mathsf{b}}_n)$. Since the sets $\mathrm{At}(W)$ and $\mathrm{At}(\overline{W})$ are disjoint, we may partition U into two (possibly empty) parts $S_{2n} := U \cap \mathrm{At}(\overline{W})$ and $T_{2n} := U \cap \mathrm{At}(W)$ which constitute the fingerprints

$$S_{2n} = \nu'_n\big(\bar{\mathsf{a}}_1, \tau^1_n(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{a}}_n, \tau^1_n(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_n, \bar{\mathsf{b}}_n)\big)$$

and

$$T_{2n} = \xi'_n\big(\bar{\mathsf{b}}_1, \tau^2_n(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{b}}_n, \tau^2_n(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_n, \bar{\mathsf{b}}_n)\big).$$

Claim II guarantees the existence of some representative $\langle \bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_n, \bar{\mathsf{d}}_n\rangle := \alpha_{n, \langle \overline{S}^{(n)}, \overline{T}^{(n)}\rangle}$ such that

$$S_{2n-1} = \nu_n\big(\bar{\mathsf{c}}_1, \tau^1_n(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{c}}_{n-1}, \tau^1_{n-1}(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{n-1}, \bar{\mathsf{d}}_{n-1}), \bar{\mathsf{c}}_n\big)$$

and

$$T_{2n-1} = \xi_n\big(\bar{\mathsf{d}}_1, \tau^2_n(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{d}}_{n-1}, \tau^2_{n-1}(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_{n-1}, \bar{\mathsf{d}}_{n-1}), \bar{\mathsf{d}}_n\big).$$

The mappings τ^1_n, τ^2_n are defined in such a way that

$$\begin{aligned} S_{2n} &= \nu'_n\big(\bar{\mathsf{a}}_1, \tau^1_n(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{a}}_n, \tau^1_n(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_n, \bar{\mathsf{b}}_n)\big) \\ &= \nu'_n\big(\bar{\mathsf{c}}_1, \sigma^1_n(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{c}}_n, \sigma^1_n(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_n, \bar{\mathsf{d}}_n)\big) \end{aligned}$$

and

$$\begin{aligned} T_{2n} &= \xi'_n\big(\bar{\mathsf{b}}_1, \tau^2_n(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathsf{b}}_n, \tau^2_n(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_n, \bar{\mathsf{b}}_n)\big) \\ &= \xi'_n\big(\bar{\mathsf{d}}_1, \sigma^2_n(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1), \ldots, \bar{\mathsf{d}}_n, \sigma^2_n(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_n, \bar{\mathsf{d}}_n), \bar{\mathsf{d}}_n\big)\ . \end{aligned}$$

Consequently, we have $U = \mathrm{out}_\tau(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_n, \bar{\mathsf{b}}_n) = \mathrm{out}_\sigma(\bar{\mathsf{c}}_1, \bar{\mathsf{d}}_1, \ldots, \bar{\mathsf{c}}_n, \bar{\mathsf{d}}_n) \in \mathrm{Out}_\sigma$.

Altogether, it follows that $\mathrm{Out}_\tau \subseteq \mathrm{Out}_\sigma$. □

Lemma 7.2.11 entails the existence of satisfying strategies that are uniform simultaneously for the fingerprint functions $\nu_k, \nu'_k, \xi_k, \xi'_k$, if there are satisfying strategies at all. Since this uniformity applies with respect to finitely many fingerprints, the existence of such strategies reveals the weak character of certain dependences in the sentence φ. The next lemma makes this explicit.

Lemma 7.2.12. *Suppose that $\mathcal{A}$ is a model of φ and let σ be a strategy that is satisfying for φ under $\mathcal{A}$. There is some sequence $m_1, \ldots, m_n$ of positive integers and there is a strategy τ and a family of mappings $\rho^j_k : \mathsf{A}^{|\bar{\mathsf{x}}^2_1|} \times \ldots \times \mathsf{A}^{|\bar{\mathsf{x}}^2_k|} \to \mathsf{A}^{|\bar{\mathsf{y}}^2_k|}$ with $1 \leq k \leq n$ and $1 \leq j \leq m_k$ such the following conditions are met.*

(a) For every k and for all sequences $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k$, $\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_k$ there is some j such that we have $\tau^2_k(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{b}}_k) = \rho^j_k(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_k)$.

(b) The strategy τ is satisfying for φ under $\mathcal{A}$.

Proof. Without loss of generality, we assume that σ is ν-ξ-uniform (cf. Requirement (a) in Lemma 7.2.11). In other words, we assume that σ is ν-ξ-uniform. Before we start constructing τ, we define the following notation. Let $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k$ and $\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_k$ be sequences with $\bar{\mathsf{a}}_i \in \mathsf{A}^{|\bar{\mathsf{x}}^1_i|}$ and $\bar{\mathsf{b}}_i \in \mathsf{A}^{|\bar{\mathsf{x}}^2_i|}$. By $\underline{\nu}_k$ we mean the pair $\langle S_1 \ldots S_{2k-1}, T_1 \ldots T_{2k-1}\rangle$ such that for every i, $1 \leq i \leq k$, we have

$$\begin{aligned} S_{2i-1} &:= \nu_k\big(\bar{\mathsf{a}}_1, \sigma^1_1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_2), \ldots, \bar{\mathsf{a}}_{k-1}, \sigma^1_{k-1}(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{k-1}, \bar{\mathsf{b}}_{k-1}), \bar{\mathsf{a}}_k\big)\ , \\ S_{2i} &:= \nu'_k\big(\bar{\mathsf{a}}_1, \sigma^1_1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_2), \ldots, \bar{\mathsf{a}}_{k-1}, \sigma^1_{k-1}(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{k-1}, \bar{\mathsf{b}}_{k-1}), \bar{\mathsf{a}}_k, \sigma^1_k(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{b}}_k)\big)\ , \\ T_{2i-1} &:= \xi_k\big(\bar{\mathsf{b}}_1, \sigma^2_1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_2), \ldots, \bar{\mathsf{b}}_{k-1}, \sigma^2_{k-1}(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{k-1}, \bar{\mathsf{b}}_{k-1}), \bar{\mathsf{b}}_k\big)\ , \\ T_{2i} &:= \xi'_k\big(\bar{\mathsf{b}}_1, \sigma^2_1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_2), \ldots, \bar{\mathsf{b}}_{k-1}, \sigma^2_{k-1}(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{k-1}, \bar{\mathsf{b}}_{k-1}), \bar{\mathsf{b}}_k, \sigma^2_k(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{b}}_k)\big)\ . \end{aligned}$$

We now show that there is a witness for the strategy τ. For every k, let m_k be the number of pairs $\langle S_1 \ldots S_{2k-1}, T_1 \ldots T_{2k-1}\rangle$ for which

$S_{2i-1} \in \mathrm{im}_\sigma(\nu_i)$ and $S_{2i} \in \mathrm{im}_\sigma(\nu'_i)$ for every i, and $S_{2k-1} \in S_{2k-2} \in \ldots \in S_1$,

$T_{2i-1} \in \mathrm{im}_\sigma(\xi_i)$ and $T_{2i} \in \mathrm{im}_\sigma(\xi'_i)$ for every i, and $T_{2k-1} \in T_{2k-2} \in \ldots \in T_1$.

We associate with each such pair $\langle S_1 \ldots S_{2k-1}, T_1 \ldots T_{2k-1}\rangle$ one mapping ρ^j_k and set ρ^j_k

$$\rho^j_k(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_k) := \sigma^2_k(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{b}}_k)$$

for all sequences $\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_k$ for which there exists a sequence $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k$ with $\underline{\nu}_k(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_k, \bar{\mathsf{b}}_k) = \langle S_1 \ldots S_{2k-1}, T_1 \ldots T_{2k-1}\rangle$. For all other $\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_k$, ρ^j_k shall be undefined. By ν-ξ-uniformity of σ, the ρ^j_k are well defined.

It is now easy to see that we can set $\tau^1_k := \sigma^1_k$ and $\tau^2_k := \sigma^2_k$ for every k. □

Now consider again the two sentences $\varphi = \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \psi$ and

$$\varphi' := \exists f_1 \ldots f_m . \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_k \exists (\bar{\mathrm{y}}_k \setminus \{y_*\}) . \bigvee_{1 \le i \le m} \forall \bar{\mathrm{x}}_{k+1} \exists \bar{\mathrm{x}}_{k+1} \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \psi\big[y_*/f_i(\bar{\mathrm{x}}_*)\big]$$

from Theorem 7.2.8, where we use $m := m_k$ with m_k being defined like in the proof of Lemma 7.2.12 as the number of pairs $\langle S_1 \ldots S_{2k-1}, T_1 \ldots T_{2k-1}\rangle$ for which we have

$S_{2i-1} \in \mathrm{im}_\sigma(\nu_i)$ and $S_{2i} \in \mathrm{im}_\sigma(\nu'_i)$ for every i, and $S_{2k-1} \in S_{2k-2} \in \ldots \in S_1$,

$T_{2i-1} \in \mathrm{im}_\sigma(\xi_i)$ and $T_{2i} \in \mathrm{im}_\sigma(\xi'_i)$ for every i, and $T_{2k-1} \in T_{2k-2} \in \ldots \in T_1$.

It is easy to see that we have $\varphi' \models \varphi$.

Let $\mathcal{A}$ be a model of φ. Then, there exists a strategy σ that is satisfying for φ under $\mathcal{A}$. Applying Lemma 7.2.12, we can assume that σ is ν-ξ-uniform. In particular, there are mappings $\rho^1, \ldots, \rho^m$ with $\rho^j : \mathsf{A}^{|\bar{\mathrm{x}}^2_1|} \times \ldots \times \mathsf{A}^{|\bar{\mathrm{x}}^2_k|} \to \bar{\mathrm{y}}^2_k$ that exhibit the following properties. For all sequences $\bar{\mathsf{a}}_1, \ldots, \bar{\mathsf{a}}_k$, $\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_k$ there is some j for which we have

$$\begin{aligned}\mathcal{A}, [&\bar{\mathrm{x}}^1_1 \mapsto \bar{\mathsf{a}}_1, \bar{\mathrm{x}}^2_1 \mapsto \bar{\mathsf{b}}_1, \ldots, \bar{\mathrm{x}}^1_k \mapsto \bar{\mathsf{a}}_k, \bar{\mathrm{x}}^2_k \mapsto \bar{\mathsf{b}}_k,\\ &\bar{\mathrm{y}}^1_1 \mapsto \sigma^1_1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathrm{y}}^1_{k-1} \mapsto \sigma^1_{k-1}(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{k-1}, \bar{\mathsf{b}}_{k-1}),\\ &\bar{\mathrm{y}}^2_1 \mapsto \sigma^2_1(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1), \ldots, \bar{\mathrm{y}}^2_{k-1} \mapsto \sigma^2_{k-1}(\bar{\mathsf{a}}_1, \bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{a}}_{k-1}, \bar{\mathsf{b}}_{k-1}),\\ &\qquad\qquad \bar{\mathrm{y}}^2_k \mapsto \rho^j(\bar{\mathsf{b}}_1, \ldots, \bar{\mathsf{b}}_k)] \;\models\; \forall \bar{\mathrm{x}}_{k+1} \exists \bar{\mathrm{y}}_{k+1} \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \psi\ .\end{aligned}$$

Since $\bar{\mathrm{x}}_* = (\bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_k) \cap W$ coincides with $\langle \bar{\mathrm{x}}^2_1, \ldots, \bar{\mathrm{x}}^2_k\rangle$ and because of $y_* \in \bar{\mathrm{y}}_k \cap W = \bar{\mathrm{y}}^2_k$, this implies that $\mathcal{A} \models \varphi'$. Therefore, we have $\varphi \models \varphi'$.

This finishes the proof of Theorem 7.2.8.

Notice that Theorem 7.2.8 does not subsume the results that we have obtained for GBSR and GAF in Chapter 4. Even in the setting of SF the theorem would not suffice to show decidability. Some dependences can still be removed and the theorem strengthened. For a more fine-grained analysis of dependences, it might make sense to define W differently — see the conjecture below. As already mentioned earlier, the set $\widehat{V}^{\trianglerighteq v}$ overapproximates the set of variables that must lie in the scope of the quantifier $\mathcal{Q}v$. For example, in the sentence $\forall u \forall v \exists y . P(u, v) \wedge R(u, y)$ we have $\widehat{V}^{\trianglerighteq u} = \{u, v, y\}$, $\widehat{V}^{\trianglerighteq v} = \{v\}$, and $\widehat{V}^{\trianglerighteq y} = \{y\}$. Evidently, y could very well be outside of the scope of $\forall v$, although $\widehat{V}^{\trianglerighteq u}$ does not indicate this; however, $\widehat{V}^{\trianglerighteq v}$ does indicate it. In contrast to the setting of Theorem 7.2.8, there is no neat partition of the set of atoms into one part that exclusively contains variables y only weakly depends on and a disjoint part collecting the variables having strong ties with y.

Compared to Theorem 7.2.7, Theorem 7.2.8 can already be considered an advance. Nevertheless, it leaves plenty of room for improvement. Using more sophisticated arguments, one should be able to proof the following strengthening.

Conjecture 7.2.13. *Let $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \psi$ be a first-order sentence in standard form with quantifier-free ψ. Consider any k, $1 \le k \le n$, and let y_* be some variable in $\bar{\mathrm{y}}_k$. Let $\bar{\mathrm{x}}_*$ be a tuple*

containing exactly the variables x from $\bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_k$ for which we have $y_* \in \widehat{V}^{\trianglerighteq x}$. Then, there is some positive integer m such that φ is equivalent to the sentence

$$\varphi' := \exists f_1 \ldots f_m. \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_k \exists (\bar{\mathrm{y}}_k \setminus \{y_*\}). \bigvee_{1 \leq i \leq m} \forall \bar{\mathrm{x}}_{k+1} \exists \bar{\mathrm{x}}_{k+1} \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n. \psi\big[y_*/f_i(\bar{\mathrm{x}}_*)\big] \ .$$

Intuitively, the conjecture states that the existentially quantified y_* cannot depend strongly on any universally quantified variable v with $y_* \notin \widehat{V}^{\trianglerighteq v}$. Moreover, in such cases we replace y_* with several, though finitely many, Skolem terms that need not have v as an argument.

Further room for improvement may be provided when taking Boolean structure into account, either similarly to the approach outlined in Section 3.6 or in a yet-to-be developed way.

7.3 Elimination of Second-Order Quantifiers in Second-Order SF

Elimination of second-order quantifiers can be conceived as a generalization of the satisfiability problem of first-order logic. Consider any relational first-order sentence φ and let $P_1, \ldots, P_n$ be an enumeration of all the predicate symbols occurring in φ. Now consider the second-order sentence $\exists P_1 \ldots P_n. \varphi$. Elimination of all second-order quantifiers in $\exists P_1 \ldots P_n. \varphi$ yields an equivalent first-order sentence ψ in which the equality sign is the only predicate symbol. In other words, ψ is an $\mathrm{MFO}_{\approx}$ sentence over the empty vocabulary. Therefore, we can decide satisfiability of ψ. In summary, any procedure that can eliminate all second-order quantifiers in $\exists P_1 \ldots P_n. \varphi$ can be turned into a decision procedure for the first-order sentence φ.

Second-order quantifier elimination has a number of applications in knowledge representation and automated reasoning, see, e.g., [GSS08, Wer15b, Wer15a, KRSW17]. It is a classical result that the monadic fragment of second-order logic (MSO) admits elimination of second-order quantifiers. This was discovered by Löwenheim [Löw15], Skolem [Sko19], and Behmann [Beh22]. As almost all of the novel fragments we introduced in Section 3 generalize MFO while retaining a decidable satisfiability problem, it is natural to ask whether second-order versions of these fragments admit elimination of second-order quantifiers. We shall only sketch a preliminary answer to this questions for the simplest fragment and, hence, focus on SF in the present section. Interestingly, already Ackermann gave a counterexample. In an article from 1935 [Ack35], Ackermann argued that the quantifier $\exists P$ in the following formula cannot be eliminated: $\exists P.\ P(x) \wedge \neg P(y) \wedge \forall uv.\ \neg P(u) \vee P(v) \vee \neg N(u, v)$. The only atom in this formula that could potentially break the separateness condition is $N(u, v)$. But since both variables u and v are universally quantified, universal variables are separated from existential variables and the sentence is in SF.

Although Ackermann's observation seems to be discouraging, it only means that there is, apparently, no straight-forward way of extending the quantifier-elimination techniques that work for MSO to a second-order version of SF. In the following we shall present certain syntactic restrictions that allow the elimination of existentially quantified unary predicate symbols in separated formulas. To emphasize it again: the presented results are of a preliminary character and are not yet fully developed. They provide only a first hint at some directions that might be worth following in future work.

Our starting point will be the equivalence-preserving transformation from SF into BSR, which we need to adapt only slightly to the needs of second-order quantifier elimination.

Lemma 7.3.1. *Let $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n. \psi(\bar{\mathrm{x}}, \bar{\mathrm{y}}, \bar{\mathrm{z}})$ be a relational first-order* formula *in standard form with quantifier-free ψ. We set $\bar{\mathrm{x}} := \bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_n$ and $\bar{\mathrm{y}} := \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_n$ and we assume that every variable in the quantifier prefix also occurs in the matrix. Let $\widetilde{\mathrm{x}}_1, \ldots, \widetilde{\mathrm{x}}_{m_1} \subseteq \bar{\mathrm{x}}$ and $\widetilde{\mathrm{y}}_1, \ldots, \widetilde{\mathrm{y}}_{m_2} \subseteq \bar{\mathrm{y}}$ be partitions of the sets $\bar{\mathrm{x}}$ and $\bar{\mathrm{y}}$, respectively, such that the $\widetilde{\mathrm{x}}_1, \ldots, \widetilde{\mathrm{x}}_{m_1}, \widetilde{\mathrm{y}}_1, \ldots, \widetilde{\mathrm{y}}_{m_2}$ are nonempty, pairwise disjoint, and pairwise separated in φ. Then, φ is equivalent to a finite disjunction of formulas of the form*

$$\Big(\bigwedge_k \forall \widetilde{\mathrm{x}}''_k. \bigvee_\ell K_{k\ell}(\widetilde{\mathrm{x}}''_k, \bar{\mathrm{z}})\Big) \ \wedge \ \Big(\bigwedge_i \exists \widetilde{\mathrm{y}}''_i. \bigwedge_j L_{ij}(\widetilde{\mathrm{y}}''_i, \bar{\mathrm{z}})\Big) \ ,$$

where the $K_{k\ell}$ and the L_{ij} are literals whose atoms are renamed variants of atoms that occur in φ. Moreover, any two sets $\widetilde{\mathrm{x}}''_{k_1}, \widetilde{\mathrm{x}}''_{k_2}$ with $k_1 \neq k_2$, $\widetilde{\mathrm{y}}''_{i_1}, \widetilde{\mathrm{y}}''_{i_2}$ with $i_1 \neq i_2$, and $\widetilde{\mathrm{x}}''_k, \widetilde{\mathrm{y}}''_i$ are separated in the resulting formula.

Proof. For convenience, we pretend that $\bar{\mathrm{z}}$ is empty. The argument works for nonempty $\bar{\mathrm{z}}$ as well.

We transform φ into an equivalent CNF formula of the form

$$\forall\bar{\mathrm{x}}_1 \exists\bar{\mathrm{y}}_1 \ldots \forall\bar{\mathrm{x}}_n \exists\bar{\mathrm{y}}_n . \bigwedge_{i\in I} \Big(\chi_i(\bar{\mathrm{x}}) \vee \bigvee_{k\in J_i} L_k(\bar{\mathrm{y}}) \Big)$$

where I and the J_i are finite, pairwise disjoint sets of indices, the subformulas χ_i are disjunctions of literals, and the L_k are literals. By Lemma 3.2.4, we can construct an equivalent formula of the form

$$\varphi' := \forall\bar{\mathrm{x}}_1 \exists\bar{\mathrm{y}}_1 \ldots \forall\bar{\mathrm{x}}_n . \bigwedge_{\substack{S\subseteq I \\ S\neq\emptyset}} \Big(\bigvee_{i\in S} \chi_i(\bar{\mathrm{x}}) \Big) \vee \bigvee_{f\in\mathcal{F}} \Big(\exists\bar{\mathrm{y}}_n . \bigwedge_{i\in S} \eta_{f(i)}(\bar{\mathrm{y}}) \Big) \qquad \varphi'$$

I, J_i, L_k

where $\mathcal{F}$ is the set of all selection functions over the index sets J_i, $i \in I$. We shift the universal quantifier block $\forall\bar{\mathrm{x}}_n$ inwards and thus obtain

$$\varphi'' := \forall\bar{\mathrm{x}}_1 \exists\bar{\mathrm{y}}_1 \ldots \exists\bar{\mathrm{y}}_{n-1} . \bigwedge_{\substack{S\subseteq I \\ S\neq\emptyset}} \Big(\forall\bar{\mathrm{x}}_n . \bigvee_{i\in S} \chi_i(\bar{\mathrm{x}}) \Big) \vee \bigvee_{f\in\mathcal{F}} \Big(\exists\bar{\mathrm{y}}_n . \bigwedge_{i\in S} \eta_{f(i)}(\bar{\mathrm{y}}) \Big) . \qquad \varphi''$$

We now iterate these two steps in an alternating fashion until all quantifier blocks have been shifted inwards in the described way. The constituents of the result $\varphi^{(3)} := \bigwedge_q \Big(\chi_q^{(3)} \vee \bigvee_p \eta_{qp}^{(3)} \Big)$ of this process have the form

$\varphi^{(3)}$

$$\chi_q^{(3)} = \forall\bar{\mathrm{x}}_1 . \bigvee_{\ell_1} \forall\bar{\mathrm{x}}_2 . \bigvee_{\ell_2} \Big(\ldots \Big(\bigvee_{\ell_{n-1}} \forall\bar{\mathrm{x}}_n . \bigvee_{i\in S_{\ell_1,\ldots,\ell_{n-1}}} \chi_i(\bar{\mathrm{x}}) \Big) \ldots \Big) \qquad \chi_q^{(3)}$$

where the $S_{\ell_1,\ldots,\ell_n-1}$ are certain subsets of I and the χ_i are still disjunctions of literals, and

$$\eta_{qp}^{(3)} = \exists\bar{\mathrm{y}}_1 . \bigwedge_{\ell_1} \exists\bar{\mathrm{y}}_2 . \bigwedge_{\ell_2} \Big(\ldots \Big(\bigwedge_{\ell_{n-1}} \exists\bar{\mathrm{y}}_n . \bigwedge_{k\in J_{\ell_1,\ldots,\ell_{n-1}}} L_k(\bar{\mathrm{y}}) \Big) \ldots \Big) \qquad \eta_{qp}^{(3)}$$

where the $J_{\ell_1,\ldots,\ell_{n-1}}$ are certain subsets of $\bigcup_{i\in I} J_i$.

By definition of the sets $\widetilde{\mathrm{x}}_1, \ldots, \widetilde{\mathrm{x}}_{m_1}$, which are pairwise separated in the $\chi_q^{(3)}$, we can rewrite every $\chi_q^{(3)}$ into the following form by regrouping the inner disjuncts:

$$\chi_q^{(4)} = \forall\bar{\mathrm{x}}_1 . \bigvee_{\ell_1} \forall\bar{\mathrm{x}}_2 . \bigvee_{\ell_2} \Big(\ldots \Big(\bigvee_{\ell_{n-1}} \forall\bar{\mathrm{x}}_n . \bigvee_{1\leq i'\leq m_1} \chi'_{\bar{\ell}i'}(\widetilde{\mathrm{x}}_{i'}) \Big) \ldots \Big) \qquad \chi_q^{(4)}$$

where the $\chi'_{\bar{\ell}i'}$ are (possibly empty) disjunctions of literals. Analogously, we rewrite every $\eta_{qp}^{(3)}$ into the form

$$\eta_{qp}^{(4)} = \exists\bar{\mathrm{y}}_1 . \bigwedge_{\ell_1} \exists\bar{\mathrm{y}}_2 . \bigwedge_{\ell_2} \Big(\ldots \Big(\bigwedge_{\ell_{n-1}} \exists\bar{\mathrm{y}}_n . \bigwedge_{1\leq j'\leq m_2} \eta'_{\bar{\ell}j'}(\widetilde{\mathrm{y}}_{j'}) \Big) \ldots \Big) \qquad \eta_{qp}^{(4)}$$

where the $\eta'_{\bar{\ell}j'}$ are (possibly empty) conjunctions of literals.

We then observe the following equivalences, starting from $\chi_q^{(4)}$:

$$\begin{aligned}
&\forall \bar{\mathrm{x}}_1. \bigvee_{\ell_1} \forall \bar{\mathrm{x}}_2. \bigvee_{\ell_2} \Big(\dots \Big(\bigvee_{\ell_{n-1}} \forall \bar{\mathrm{x}}_n. \bigvee_{1 \leq i' \leq m_1} \chi'_{\bar{\ell} i'}(\widetilde{\mathrm{x}}_{i'}) \Big) \dots \Big) \\
&\eqsim \forall \bar{\mathrm{x}}_1. \bigvee_{\ell_1} \forall \bar{\mathrm{x}}_2. \bigvee_{\ell_2} \Big(\dots \Big(\bigvee_{\ell_{n-1}} \bigvee_{1 \leq i' \leq m_1} \forall (\bar{\mathrm{x}}_n \cap \widetilde{\mathrm{x}}_{i'}).\ \chi'_{\bar{\ell} i'}(\widetilde{\mathrm{x}}_{i'}) \Big) \dots \Big) \\
&\eqsim \forall \bar{\mathrm{x}}_1. \bigvee_{\ell_1} \forall \bar{\mathrm{x}}_2. \bigvee_{\ell_2} \Big(\dots \Big(\bigvee_{1 \leq i' \leq m_1} \bigvee_{\ell'_{n-1}} \forall (\bar{\mathrm{x}}_n \cap \widetilde{\mathrm{x}}_{i'}).\ \chi'_{\bar{\ell}' i'}(\widetilde{\mathrm{x}}_{i'}) \Big) \dots \Big) \\
&\qquad \vdots \\
&\eqsim \bigvee_{1 \leq i' \leq m_1} \forall (\bar{\mathrm{x}}_1 \cap \widetilde{\mathrm{x}}_{i'}). \bigvee_{\ell'_1} \forall (\bar{\mathrm{x}}_2 \cap \widetilde{\mathrm{x}}_{i'}). \bigvee_{\ell'_2} \Big(\dots \Big(\bigvee_{\ell'_{n-1}} \forall (\bar{\mathrm{x}}_n \cap \widetilde{\mathrm{x}}_{i'}).\ \chi'_{\bar{\ell}' i'}(\widetilde{\mathrm{x}}_{i'}) \Big) \dots \Big) \\
&\eqsim \bigvee_{1 \leq i' \leq m_1} \forall \widetilde{\mathrm{x}}'_{i'}.\ \chi''_{i'}(\widetilde{\mathrm{x}}'_{i'}) \ ,
\end{aligned}$$

where the $\chi''_{i'}$ are disjunctions of literals. Before shifting universal quantifiers outwards in the last step of the above transformation, bound variables are renamed so that all quantifiers bind pairwise distinct variables. Analogously, we have

$$\eta_{qp}^{(4)} \eqsim \bigwedge_{1 \leq j' \leq m_2} \exists \widetilde{\mathrm{y}}'_{j'}.\ \eta''_{j'}(\widetilde{\mathrm{y}}'_{j'}) \ ,$$

where the $\eta''_{j'}$ are conjunctions of literals.

Consequently, we have rewritten $\varphi^{(3)} = \bigwedge_q \Big(\chi_q^{(3)} \vee \bigvee_p \eta_{qp}^{(3)} \Big)$ into an equivalent formula $\varphi^{(4)}$ of the form

$\varphi^{(4)}$

$$\varphi^{(4)} = \bigwedge_q \Big(\Big(\bigvee_{1 \leq i' \leq m_1} \forall \widetilde{\mathrm{x}}'_{i'}.\ \chi''_{qi'}(\widetilde{\mathrm{x}}'_{i'}) \Big) \vee \Big(\bigvee_p \bigwedge_{1 \leq j' \leq m_2} \exists \widetilde{\mathrm{y}}'_{j'}.\ \eta''_{qpj'}(\widetilde{\mathrm{y}}'_{j'}) \Big) \Big) \ .$$

After renaming bound variables again in such a way that all quantifiers bind pairwise distinct variables, we transform $\varphi^{(4)}$ into an equivalent formula that is a disjunction of formulas of the form

$$\bigwedge_k \Big(\forall \widetilde{\mathrm{x}}''_k.\ \bigvee_\ell K_{k\ell}(\widetilde{\mathrm{x}}''_k) \Big) \ \wedge \ \bigwedge_i \Big(\exists \widetilde{\mathrm{y}}''_i.\ \bigwedge_j L_{ij}(\widetilde{\mathrm{y}}''_i) \Big) \ .$$

□

Lemma 7.3.1 provides the syntactic transformations for eliminating second-order quantifiers that occur in a separated formula under certain conditions.

Example 7.3.2. *Consider the SF sentence* $\varphi := \forall x_1 \exists y_1 \forall x_2 \exists y_2.\, R(x_1, x_2) \leftrightarrow Q(y_1, y_2)$. *Nested alternating quantifiers can be transformed away, as indicated by Lemma 7.3.1. An intermediate result of this process is*

$$\begin{aligned}
\forall x_1 \exists y_1. \Big(\big(\forall x_2.\, R(x_1, x_2) \big) \vee \big(\exists y_2.\, \neg Q(y_1, y_2) \big) \Big) \\
\wedge \Big(\big(\forall x_2.\, \neg R(x_1, x_2) \big) \vee \big(\exists y_2.\, Q(y_1, y_2) \big) \Big) \ .
\end{aligned}$$

Continuing the transformation, we eventually obtain

$$\begin{aligned}
&\Big(\exists y_1 y_2 y_3.\, Q(y_1, y_2) \wedge \neg Q(y_1, y_3) \Big) \\
&\vee \Big(\big(\forall x_1 x_2.\, R(x_1, x_2) \big) \wedge \big(\exists y_1 y_2.\, Q(y_1, y_2) \big) \Big) \\
&\vee \Big(\big(\forall x_1 x_2.\, \neg R(x_1, x_2) \big) \wedge \big(\exists y_1 y_2.\, \neg Q(y_1, y_2) \big) \Big) \\
&\vee \Big(\big(\forall x_1 x_2 x_3.\, R(x_1, x_2) \vee \neg R(x_1, x_3) \big) \wedge \big(\exists y_1 y_2.\, Q(y_1, y_2) \big) \wedge \big(\exists y_3 y_4.\, \neg Q(y_3, y_4) \big) \Big) \ ,
\end{aligned}$$

which is equivalent to φ *but does not contain any quantifier alternation.*

Next, we formulate syntactic restrictions that enable the elimination of second-order quantifiers over unary predicates from sentences that belong to a second-order variant of SF. Again, separateness of sets of variables in a formula plays a central role in our criterion. However, this time it is not only of interest that universally quantified and existentially quantified variables are separated. In addition, it is important that within each set of non-separated variables there is at most one variable that occurs as the argument of the predicate symbol that is bound by the second-order quantifier we intend to eliminate.

Lemma 7.3.3. *Let $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \psi(\bar{\mathrm{x}}, \bar{\mathrm{y}}, \bar{\mathrm{z}})$ be a relational first-order formula in which ψ is quantifier free and the sets $\bar{\mathrm{x}} := \bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_n$ and $\bar{\mathrm{y}} := \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_n$ are separated. We assume that every variable occurring in the quantifier prefix and in $\bar{\mathrm{z}}$ also occurs in the matrix ψ.*

Let $\widetilde{\mathrm{x}}_1, \ldots, \widetilde{\mathrm{x}}_{m_1} \subseteq \bar{\mathrm{x}}$ and $\widetilde{\mathrm{y}}_1, \ldots, \widetilde{\mathrm{y}}_{m_2} \subseteq \bar{\mathrm{y}}$ be partitions of the sets $\bar{\mathrm{x}}$ and $\bar{\mathrm{y}}$, respectively, such that the $\widetilde{\mathrm{x}}_1, \ldots, \widetilde{\mathrm{x}}_{m_1}, \widetilde{\mathrm{y}}_1, \ldots, \widetilde{\mathrm{y}}_{m_2}$ are nonempty, pairwise disjoint, and pairwise separated in φ. Let P be a unary predicate symbol satisfying the following conditions:

(1) For every set $\widetilde{\mathrm{x}}_i$, $1 \leq i \leq m_1$, there is at most one variable $x_i^* \in \widetilde{\mathrm{x}}_i$ for which φ contains atoms $P(x_i^*)$.

(2) For every set $\widetilde{\mathrm{y}}_i$, $1 \leq i \leq m_2$, there is at most one variable $y_i^* \in \widetilde{\mathrm{y}}_i$ for which φ contains atoms $P(y_i^*)$.

Then, $\exists P.\, \varphi$ is equivalent to a finite disjunction of formulas of the form

$$\begin{aligned} \theta(\bar{\mathrm{z}}) \;\wedge\; \exists P.\; & \bigwedge_{k_1} \Big(\forall \widetilde{\mathrm{x}}'_{k_1} .\; \chi_{k_1}(\widetilde{\mathrm{x}}'_{k_1}, \bar{\mathrm{z}}) \vee P(x^*_{k_1}) \Big) \wedge \bigwedge_{k_2} \Big(\forall \widetilde{\mathrm{x}}'_{k_2} .\; \chi'_{k_2}(\widetilde{\mathrm{x}}'_{k_2}, \bar{\mathrm{z}}) \vee \neg P(x^*_{k_2}) \Big) \\ & \wedge \bigwedge_{i_1} \Big(\exists \widetilde{\mathrm{y}}'_{i_1} .\; \eta_{i_1}(\widetilde{\mathrm{y}}'_{i_1}, \bar{\mathrm{z}}) \wedge P(y^*_{i_1}) \Big) \wedge \bigwedge_{i_2} \Big(\exists \widetilde{\mathrm{y}}'_{i_2} .\; \eta'_{i_2}(\widetilde{\mathrm{y}}'_{i_2}, \bar{\mathrm{z}}) \wedge \neg P(y^*_{i_2}) \Big) \\ & \wedge \bigwedge_{\ell_1} P(z^*_{\ell_1}) \wedge \bigwedge_{\ell_2} \neg P(z^*_{\ell_2}) \;, \end{aligned}$$

where (a) the χ_{k_1} and the χ'_{k_2} are disjunctions of literals and the η_{i_1} and the η'_{i_2} are conjunctions of literals, (b) all the atoms in θ and in the χ_{k_1}, χ'_{k_2}, η_{i_1}, and η'_{i_2} are renamed variants of atoms that occur in φ and do not contain the predicate symbol P, and (c) the variables $z^*_{\ell_1}$, $z^*_{\ell_2}$ are pairwise distinct and stem from $\bar{\mathrm{z}}$.

Proof. By Lemma 7.3.1, we know that φ can be rewritten into an equivalent formula that is a finite disjunction of formulas in which no universal quantifier lies within the scope of an existential quantifier and vice versa. We apply this transformation to φ and obtain a formula as described in Lemma 7.3.1. In the next step, we isolate atoms that exclusively contain variables from $\bar{\mathrm{z}}$, shift first-order quantifiers inwards so that these atoms are not within their scopes anymore, and transform the result into a formula φ' that is a disjunction of formulas of the form

$$\Big(\bigwedge_k \forall \widetilde{\mathrm{x}}'_k .\; \bigvee_\ell K_{k\ell}(\widetilde{\mathrm{x}}'_k, \bar{\mathrm{z}}) \Big) \;\wedge\; \Big(\bigwedge_i \exists \widetilde{\mathrm{y}}'_i .\; \bigwedge_j L_{ij}(\widetilde{\mathrm{y}}'_i, \bar{\mathrm{z}}) \Big) \;\wedge\; \bigwedge_r M_r(\bar{\mathrm{z}}) \;,$$

where the $K_{k\ell}$ and the L_{ij} are literals whose atoms are renamed variants of atoms from φ and contain at least one variable from some $\widetilde{\mathrm{x}}'_k$ or $\widetilde{\mathrm{y}}'_i$. The M_r are literals whose atoms occur in φ and contain exclusively variables from $\bar{\mathrm{z}}$. Moreover, any two sets $\widetilde{\mathrm{x}}'_{k_1}, \widetilde{\mathrm{x}}'_{k_2}$ with $k_1 \neq k_2$, $\widetilde{\mathrm{y}}'_{i_1}, \widetilde{\mathrm{y}}'_{i_2}$ with $i_1 \neq i_2$, and $\widetilde{\mathrm{x}}'_k, \widetilde{\mathrm{y}}'_i$ are separated in φ'. By inspection of the transformations performed in the proof of Lemma 7.3.1, we observe that Conditions (1) and (2) are preserved such that they also apply to the sets $\widetilde{\mathrm{x}}'_k$ and $\widetilde{\mathrm{y}}'_i$ with respect to variables x^*_k and y^*_i, respectively.

This enables us to regroup the disjunctions and conjunctions in the constituents of φ' so that

each of these disjuncts has the form

$$\begin{aligned}
&\bigwedge_{k'}\Big(\forall\widetilde{\mathrm{x}}'_{k'}.\Big(\bigvee_{\ell'} K_{k'\ell'}(\widetilde{\mathrm{x}}'_{k'},\bar{\mathrm{z}})\Big) \vee [\neg]P(x^*_{k'})\Big)\\
\wedge\ &\bigwedge_{i'}\Big(\exists\widetilde{\mathrm{y}}'_{i'}.\Big(\bigwedge_{j'} L_{i'j'}(\widetilde{\mathrm{y}}'_{i'},\bar{\mathrm{z}})\Big) \wedge [\neg]P(y^*_{i'})\Big)\\
\wedge\ &\Big(\bigwedge_{r'} M_{r'}(\bar{\mathrm{z}})\Big) \wedge \bigwedge_{q}[\neg]P(z^*_q)\ ,
\end{aligned}$$

where the literals $K_{k'\ell'}$, $L_{i'j'}$, and $M_{r'}$ do not contain the predicate symbol P. The variables z^*_q stem from $\bar{\mathrm{z}}$. Moreover, we replace disjuncts (conjuncts) which contain two literals $P(v)$ and $\neg P(v)$ with the logical constant `true` (`false`). Having this, it only remains to regroup conjuncts and distribute the second-order quantifier $\exists P$ over the topmost disjunction, in order to obtain the formula advertised in the lemma. □

The formula resulting from Lemma 7.3.3 gives us the right starting point for the elimination of the second-order quantifier $\exists P$. Before we elaborate on this, we present the Basic Elimination Lemma that we shall employ for elimination.

Proposition 7.3.4 (Basic Elimination Lemma, see [Wer15a], Lemma 3, and [Beh22])**.** *Let P be a unary predicate symbol and let χ, η be first-order formulas in which P does not occur. Then, $\exists P.\ \big(\forall x.\,\chi \vee P(x)\big) \wedge \big(\forall x.\,\eta \vee \neg P(x)\big)$ is semantically equivalent to $\forall x.\,\chi \vee \eta$.*

Consider any formula $\varphi := \forall\bar{\mathrm{x}}_1\exists\bar{\mathrm{y}}_1 \ldots \forall\bar{\mathrm{x}}_n\exists\bar{\mathrm{y}}_n.\,\psi(\bar{\mathrm{x}},\bar{\mathrm{y}},\bar{\mathrm{z}})$ satisfying the prerequisites of Lemma 7.3.3. Moreover, let there be sets $\widetilde{\mathrm{x}}_1,\ldots,\widetilde{\mathrm{x}}_{m_1}$ and $\widetilde{\mathrm{y}}_1,\ldots,\widetilde{\mathrm{y}}_{m_2}$ and a unary predicate symbol P as described in the lemma. Then, Lemma 7.3.3 stipulates the existence of a formula equivalent to φ that is a disjunction of formulas of the form

$$\begin{aligned}
\theta(\bar{\mathrm{z}})\ \wedge\ \exists P.\ &\bigwedge_{k_1}\Big(\forall\widetilde{\mathrm{x}}'_{k_1}.\,\chi_{k_1}(\widetilde{\mathrm{x}}'_{k_1},\bar{\mathrm{z}}) \vee P(x^*_{k_1})\Big) \wedge \bigwedge_{k_2}\Big(\forall\widetilde{\mathrm{x}}'_{k_2}.\,\chi'_{k_2}(\widetilde{\mathrm{x}}'_{k_2},\bar{\mathrm{z}}) \vee \neg P(x^*_{k_2})\Big)\\
&\wedge \bigwedge_{i_1}\Big(\exists\widetilde{\mathrm{y}}'_{i_1}.\,\eta_{i_1}(\widetilde{\mathrm{y}}'_{i_1},\bar{\mathrm{z}}) \wedge P(y^*_{i_1})\Big) \wedge \bigwedge_{i_2}\Big(\exists\widetilde{\mathrm{y}}'_{i_2}.\,\eta'_{i_2}(\widetilde{\mathrm{y}}'_{i_2},\bar{\mathrm{z}}) \wedge \neg P(y^*_{i_2})\Big)\\
&\wedge \bigwedge_{\ell_1} P(z^*_{\ell_1}) \wedge \bigwedge_{\ell_2}\neg P(z^*_{\ell_2})\ ,
\end{aligned}$$

in which we can eliminate the quantifier $\exists P$ as follows. The shape of the above formula is very similar to what Behmann called "*Eliminationshauptform*" in [Beh22] (see [Wer15a] for a modern exposition of Behmann's results related to quantifier elimination). With the next two transformation steps we come closer to the syntactic shape of the "Eliminationshauptform". First, we shift the first-order quantifiers inwards that do not bind variables x^*_k or y^*_i:

$$\begin{aligned}
\theta(\bar{\mathrm{z}})\ \wedge\ \exists P.\ &\bigwedge_{k_1}\Big(\forall x^*_{k_1}.\ \underbrace{\big(\forall(\widetilde{\mathrm{x}}'_{k_1}\setminus\{x^*_{k_1}\}).\,\chi_{k_1}(\widetilde{\mathrm{x}}'_{k_1},\bar{\mathrm{z}})\big)}_{=:\,\chi^*_{k_1}} \vee P(x^*_{k_1})\Big)\\
&\wedge\ \bigwedge_{k_2}\Big(\forall x^*_{k_2}.\ \underbrace{\big(\forall(\widetilde{\mathrm{x}}'_{k_2}\setminus\{x^*_{k_2}\}).\,\chi_{k_2}(\widetilde{\mathrm{x}}'_{k_2},\bar{\mathrm{z}})\big)}_{=:\,\chi^*_{k_2}} \vee \neg P(x^*_{k_2})\Big)\\
&\wedge\ \bigwedge_{i_1}\Big(\exists y^*_{i_1}.\ \underbrace{\big(\exists(\widetilde{\mathrm{y}}'_{i_1}\setminus\{y^*_{i_1}\}).\,\eta'_{i_1}(\widetilde{\mathrm{y}}'_{i_1},\bar{\mathrm{z}})\big)}_{=:\,\eta^*_{i_1}} \wedge P(y^*_{i_1})\Big)\\
&\wedge\ \bigwedge_{i_2}\Big(\exists y^*_{i_2}.\ \underbrace{\big(\exists(\widetilde{\mathrm{y}}'_{i_2}\setminus\{y^*_{i_2}\}).\,\eta'_{i_2}(\widetilde{\mathrm{y}}'_{i_2},\bar{\mathrm{z}})\big)}_{=:\,\eta^*_{i_2}} \wedge \neg P(y^*_{i_2})\Big)\\
&\wedge \bigwedge_{\ell_1} P(z^*_{\ell_1}) \wedge \bigwedge_{\ell_2}\neg P(z^*_{\ell_2})\ .
\end{aligned}$$

Next, we treat the subformulas χ_k^* and η_i^* as indivisible units, shift universal quantifiers outwards (over $\bigwedge_{k_1}$ and $\bigwedge_{k_2}$) that occur in different conjuncts (and merge them while doing so), shift first-order existential quantifiers outwards (over $\bigwedge_{i_1}$ and $\bigwedge_{i_2}$; without merging them), and rename the variables that are bound by the shifted quantifiers. Moreover, we reorder the conjunctions in the scope of the quantifier blocks $\exists \bar{\mathrm{u}}$ and $\exists \bar{\mathrm{v}}$:

$$\begin{aligned}
\theta(\bar{\mathrm{z}}) \;\wedge\; \exists P.\, \Big(\forall x.\, \underbrace{\Big(\bigwedge_{k_1} \chi_{k_1}^*\big[x_{k_1}^*/x\big]\Big)}_{=:\,\chi_1^*(x,\bar{\mathrm{z}})} \vee P(x)\Big) \\
\wedge\; \Big(\forall x.\, \underbrace{\Big(\bigwedge_{k_2} \chi_{k_2}^*\big[x_{k_2}^*/x\big]\Big)}_{=:\,\chi_2^*(x,\bar{\mathrm{z}})} \vee \neg P(x)\Big) \\
\wedge\; \Big(\exists \bar{\mathrm{u}}.\, \underbrace{\Big(\bigwedge_{i_1} \eta_{i_1}^*\big[y_{i_1}^*/u_{i_1}\big]\Big)}_{=:\,\eta_1^*(\bar{\mathrm{u}},\bar{\mathrm{z}})} \wedge \bigwedge_{i_1} P(u_{i_1})\Big) \\
\wedge\; \Big(\exists \bar{\mathrm{v}}.\, \underbrace{\Big(\bigwedge_{i_2} \eta_{i_2}^*\big[y_{i_2}^*/v_{i_2}\big]\Big)}_{=:\,\eta_2^*(\bar{\mathrm{v}},\bar{\mathrm{z}})} \wedge \bigwedge_{i_2} \neg P(v_{i_2})\Big) \\
\wedge\; \Big(\bigwedge_{\ell_1} P(z_{\ell_1}^*) \wedge \bigwedge_{\ell_2} \neg P(z_{\ell_2}^*)\Big)\,.
\end{aligned}$$

In what follows we treat the χ_1^*, χ_2^* and η_1^*, η_2^* as indivisible units. One more step remains to establish a kind of "Eliminationshauptform". We shift the quantifier blocks $\exists \bar{\mathrm{u}}$ and $\exists \bar{\mathrm{v}}$ outwards over the second-order quantifier $\exists P$, reorder the conjuncts within the scope of $\exists P$, and shift $\exists P$ inwards so that its scope does not contain the η_1^*, η_2^* anymore. Moreover, we make use of Proposition 3.3.6 and turn the literals $P(u_{i_1})$ into subformulas $\forall x.\, x \approx u_{i_1} \to P(x)$. We proceed analogously with the literals $\neg P(v_{i_2})$, $P(z_{\ell_1}^*)$, and $\neg P(z_{\ell_2}^*)$:

$$\begin{aligned}
\theta(\bar{\mathrm{z}}) \;\wedge\; \exists \bar{\mathrm{u}} \bar{\mathrm{v}}.\, \eta_1^*(\bar{\mathrm{u}},\bar{\mathrm{z}}) \wedge \eta_2^*(\bar{\mathrm{v}},\bar{\mathrm{z}}) \\
\wedge \exists P.\, \Big(\forall x.\, \chi_1^*(x,\bar{\mathrm{z}}) \vee P(x)\Big) \;\wedge\; \Big(\forall x.\, \chi_2^*(x,\bar{\mathrm{z}}) \vee \neg P(x)\Big) \\
\wedge\; \Big(\forall x.\, \bigwedge_{i_1} (x \approx u_{i_1} \to P(x))\Big) \;\wedge\; \Big(\forall x.\, \bigwedge_{i_2} (x \approx v_{i_2} \to \neg P(x))\Big) \\
\wedge\; \Big(\forall x.\, \bigwedge_{\ell_1} (x \approx z_{\ell_1}^* \to P(x))\Big) \;\wedge\; \Big(\forall x.\, \bigwedge_{\ell_2} (x \approx z_{\ell_2}^* \to \neg P(x))\Big)\,.
\end{aligned}$$

At this point, the subformula staring with $\exists P$ is almost in "Eliminationshauptform". After converting the implications into disjunctions and factoring out the $[\neg]P(x)$, we arrive at a formula from which the second-order quantifier $\exists P$ can be eliminated immediately via the basic elimination lemma:

$$\begin{aligned}
\theta(\bar{\mathrm{z}}) \;\wedge\; \exists \bar{\mathrm{u}} \bar{\mathrm{v}}.\, \eta_1^*(\bar{\mathrm{u}},\bar{\mathrm{z}}) \wedge \eta_2^*(\bar{\mathrm{v}},\bar{\mathrm{z}}) \\
\wedge \exists P.\, \Big(\forall x.\, \big(\chi_1^*(x,\bar{\mathrm{z}}) \wedge \bigwedge_{i_1} x \not\approx u_{i_1} \wedge \bigwedge_{\ell_1} x \not\approx z_{\ell_1}^*\big) \vee P(x)\Big) \\
\wedge\; \Big(\forall x.\, \big(\chi_2^*(x,\bar{\mathrm{z}}) \wedge \bigwedge_{i_2} x \not\approx v_{i_2} \wedge \bigwedge_{\ell_2} x \not\approx z_{\ell_2}^*\big) \vee \neg P(x)\Big)\,.
\end{aligned}$$

Using Proposition 7.3.4, we eliminate the quantifier $\exists P$ and obtain

$$\begin{aligned}\theta(\bar{\mathrm{z}}) \;\wedge\; \exists \bar{\mathrm{u}}\bar{\mathrm{v}}.\; & \eta_1^*(\bar{\mathrm{u}},\bar{\mathrm{z}}) \wedge \eta_2^*(\bar{\mathrm{v}},\bar{\mathrm{z}}) \\ & \wedge \forall x.\, \Big(\big(\chi_1^*(x,\bar{\mathrm{z}}) \wedge \bigwedge_{i_1} x \not\approx u_{i_1} \wedge \bigwedge_{\ell_1} x \not\approx z^*_{\ell_1}\big) \\ & \qquad \vee \big(\chi_2^*(x,\bar{\mathrm{z}}) \wedge \bigwedge_{i_2} x \not\approx v_{i_2} \wedge \bigwedge_{\ell_2} x \not\approx z^*_{\ell_2}\big)\Big)\,.\end{aligned}$$

In order to convert this result into a somewhat nicer form, we proceed as described in the proof of Lemma 19 in [Wer15a]. In particular, we remove the disequations $x \not\approx y$, where x is a universally quantified variable. To this end, we first distribute disjunction over conjunction within the scope of the quantifier $\forall x$:

$$\begin{aligned}\theta(\bar{\mathrm{z}}) \;\wedge\; \exists \bar{\mathrm{u}}\bar{\mathrm{v}}.\; & \eta_1^*(\bar{\mathrm{u}},\bar{\mathrm{z}}) \wedge \eta_2^*(\bar{\mathrm{v}},\bar{\mathrm{z}}) \\ & \wedge\; \forall x.\, \Big(\chi_1^*(x,\bar{\mathrm{z}}) \vee \chi_2^*(x,\bar{\mathrm{z}})\Big) \\ & \qquad \wedge\; \Big(\big(\bigwedge_{i_2} x \not\approx v_{i_2} \wedge \bigwedge_{\ell_2} x \not\approx z^*_{\ell_2}\big) \vee \chi_1^*(x,\bar{\mathrm{z}})\Big) \\ & \qquad \wedge\; \Big(\big(\bigwedge_{i_1} x \not\approx u_{i_1} \wedge \bigwedge_{\ell_1} x \not\approx z^*_{\ell_1}\big) \vee \chi_2^*(x,\bar{\mathrm{z}})\Big) \\ & \qquad \wedge\; \Big(\big(\bigwedge_{i_1} x \not\approx u_{i_1} \wedge \bigwedge_{\ell_1} x \not\approx z^*_{\ell_1}\big) \vee \big(\bigwedge_{i_2} x \not\approx v_{i_2} \wedge \bigwedge_{\ell_2} x \not\approx z^*_{\ell_2}\big)\Big)\,.\end{aligned}$$

Next, we factor the subformulas χ_1^*, χ_2^*, and $\bigwedge_{i_2} x \not\approx v_{i_2} \wedge \bigwedge_{\ell_2} x \not\approx z^*_{\ell_2}$ into the conjunctions with which they are disjunctively connected, respectively. Moreover, we turn the resulting disjunctions into implications:

$$\begin{aligned}\theta(\bar{\mathrm{z}}) \;\wedge\; \exists \bar{\mathrm{u}}\bar{\mathrm{v}}.\; & \eta_1^*(\bar{\mathrm{u}},\bar{\mathrm{z}}) \wedge \eta_2^*(\bar{\mathrm{v}},\bar{\mathrm{z}}) \\ & \wedge\; \forall x.\, \Big(\chi_1^*(x,\bar{\mathrm{z}}) \vee \chi_2^*(x,\bar{\mathrm{z}})\Big) \\ & \qquad \wedge\; \Big(\bigwedge_{i_2} \big(x \approx v_{i_2} \to \chi_1^*(x,\bar{\mathrm{z}})\big) \wedge \bigwedge_{\ell_2} \big(x \approx z^*_{\ell_2} \to \chi_1^*(x,\bar{\mathrm{z}})\big)\Big) \\ & \qquad \wedge\; \Big(\bigwedge_{i_1} \big(x \approx u_{i_1} \to \chi_2^*(x,\bar{\mathrm{z}})\big) \wedge \bigwedge_{\ell_1} \big(x \approx z^*_{\ell_1} \to \chi_2^*(x,\bar{\mathrm{z}})\big)\Big) \\ & \qquad \wedge\; \Big(\bigwedge_{i_1} \big(x \not\approx u_{i_1} \to \big(\bigwedge_{i_2} x \not\approx v_{i_2} \wedge \bigwedge_{\ell_2} x \not\approx z^*_{\ell_2}\big)\big)\Big) \\ & \qquad \wedge\; \Big(\bigwedge_{\ell_1} \big(x \not\approx z^*_{\ell_1} \to \big(\bigwedge_{i_2} x \not\approx v_{i_2} \wedge \bigwedge_{\ell_2} x \not\approx z^*_{\ell_2}\big)\big)\Big)\,.\end{aligned}$$

Finally, we apply Proposition 3.3.6 in a reverse fashion to remove the universal variable x from some of the subformulas:

$$\begin{aligned}& \theta(\bar{\mathrm{z}}) \wedge \big(\forall x.\, \chi_1^*(x,\bar{\mathrm{z}}) \vee \chi_2^*(x,\bar{\mathrm{z}})\big) \\ & \wedge \exists \bar{\mathrm{u}}\bar{\mathrm{v}}.\; \eta_1^*(\bar{\mathrm{u}},\bar{\mathrm{z}}) \wedge \eta_2^*(\bar{\mathrm{v}},\bar{\mathrm{z}}) \\ & \qquad \wedge \bigwedge_{i_2} \chi_1^*(v_{i_2},\bar{\mathrm{z}}) \;\wedge\; \bigwedge_{\ell_2} \chi_1^*(z^*_{\ell_2},\bar{\mathrm{z}}) \;\wedge\; \bigwedge_{i_1} \chi_2^*(u_{i_1},\bar{\mathrm{z}}) \;\wedge\; \bigwedge_{\ell_1} \chi_2^*(z^*_{\ell_1},\bar{\mathrm{z}}) \\ & \qquad \wedge \bigwedge_{i_1}\bigwedge_{i_2} u_{i_1} \not\approx v_{i_2} \;\wedge\; \bigwedge_{i_1}\bigwedge_{\ell_2} u_{i_1} \not\approx z^*_{\ell_2} \;\wedge\; \bigwedge_{\ell_1}\bigwedge_{i_2} z^*_{\ell_1} \not\approx v_{i_2} \;\wedge\; \bigwedge_{\ell_1}\bigwedge_{\ell_2} z^*_{\ell_1} \not\approx z^*_{\ell_2}\,.\end{aligned}$$

Consequently, we get the following result.

Theorem 7.3.5. *Let $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \psi(\bar{\mathrm{x}}, \bar{\mathrm{y}}, \bar{\mathrm{z}})$ be a relational first-order formula in which ψ is quantifier free and the sets $\bar{\mathrm{x}} := \bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_n$ and $\bar{\mathrm{y}} := \bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_n$ are separated. We assume that every variable occurring in the quantifier prefix and in $\bar{\mathrm{z}}$ also occurs in the matrix ψ.*

Let $\widetilde{\mathrm{x}}_1, \ldots, \widetilde{\mathrm{x}}_{m_1} \subseteq \bar{\mathrm{x}}$ and $\widetilde{\mathrm{y}}_1, \ldots, \widetilde{\mathrm{y}}_{m_2} \subseteq \bar{\mathrm{y}}$ be partitions of the sets $\bar{\mathrm{x}}$ and $\bar{\mathrm{y}}$, respectively, such that the $\widetilde{\mathrm{x}}_1, \ldots, \widetilde{\mathrm{x}}_{m_1}, \widetilde{\mathrm{y}}_1, \ldots, \widetilde{\mathrm{y}}_{m_2}$ are nonempty, pairwise disjoint, and pairwise separated in φ. Let P be a unary predicate symbol satisfying the following conditions:

(1) For every set $\widetilde{\mathrm{x}}_i$, $1 \leq i \leq m_1$, there is at most one variable $x_i^* \in \widetilde{\mathrm{x}}_i$ for which φ contains atoms $P(x_i^*)$.

(2) For every set $\widetilde{\mathrm{y}}_i$, $1 \leq i \leq m_2$, there is at most one variable $y_i^* \in \widetilde{\mathrm{y}}_i$ for which φ contains atoms $P(y_i^*)$.

Then, $\exists P. \varphi$ is equivalent to some first-order formula φ' that is a finite disjunction of formulas of the form

$$
\begin{aligned}
&\theta(\bar{\mathrm{z}}) \wedge \big(\forall x. \chi_1^*(x, \bar{\mathrm{z}}) \vee \chi_2^*(x, \bar{\mathrm{z}})\big) \\
&\wedge\ \exists \bar{\mathrm{u}} \bar{\mathrm{v}}. \eta_1^*(\bar{\mathrm{u}}, \bar{\mathrm{z}}) \wedge \eta_2^*(\bar{\mathrm{v}}, \bar{\mathrm{z}}) \\
&\qquad \wedge \bigwedge_{i_2} \chi_1^*(v_{i_2}, \bar{\mathrm{z}}) \ \wedge \bigwedge_{\ell_2} \chi_1^*(z_{\ell_2}^*, \bar{\mathrm{z}}) \ \wedge \bigwedge_{i_1} \chi_2^*(u_{i_1}, \bar{\mathrm{z}}) \ \wedge \bigwedge_{\ell_1} \chi_2^*(z_{\ell_1}^*, \bar{\mathrm{z}}) \\
&\qquad \wedge \bigwedge_{i_1} \bigwedge_{i_2} u_{i_1} \not\approx v_{i_2} \ \wedge \bigwedge_{i_1} \bigwedge_{\ell_2} u_{i_1} \not\approx z_{\ell_2}^* \ \wedge \bigwedge_{\ell_1} \bigwedge_{i_2} z_{\ell_1}^* \not\approx v_{i_2} \ \wedge \bigwedge_{\ell_1} \bigwedge_{\ell_2} z_{\ell_1}^* \not\approx z_{\ell_2}^* \ .
\end{aligned}
$$

Moreover, all the u_{i_1} are variables from $\bar{\mathrm{u}}$, the v_{i_2} are from $\bar{\mathrm{v}}$, and the $z_{\ell_1}^*$ and $z_{\ell_2}^*$ are free variables from $\bar{\mathrm{z}}$.

Example 7.3.6. *Consider the sentence $\varphi := \exists P. \forall x_1 \exists y \forall x_2. R(x_1, x_2) \leftrightarrow P(y)$. We transform it into the equivalent sentence*

$$
\begin{aligned}
\exists P. \Big(&\forall x_1 x_2 x_3. R(x_1, x_2) \vee \neg R(x_1, x_3)\Big) \\
&\wedge \Big(\big(\forall x_1 x_2. R(x_1, x_2)\big) \vee \big(\exists y. \neg P(y)\big)\Big) \\
&\wedge \Big(\big(\forall x_1 x_2. \neg R(x_1, x_2)\big) \vee \big(\exists y. P(y)\big)\Big) \ .
\end{aligned}
$$

For the sake of simplicity, we shift $\exists P$ inwards so that its scope only stretches over the last two conjuncts, which we thereafter transform into a disjunction of conjunctions. This yields

$$
\begin{aligned}
&\Big(\forall x_1 x_2 x_3. R(x_1, x_2) \vee \neg R(x_1, x_3)\Big) \\
&\wedge\ \Big(\exists P. \Big(\big(\forall x_1 x_2. R(x_1, x_2)\big) \wedge \big(\exists y. P(y)\big)\Big) \\
&\qquad \vee \Big(\big(\forall x_1 x_2. \neg R(x_1, x_2)\big) \wedge \big(\exists y. \neg P(y)\big)\Big) \\
&\qquad \vee \Big(\big(\exists y. P(y)\big) \wedge \big(\exists y. \neg P(y)\big)\Big)\Big) \ .
\end{aligned}
$$

Since we can distribute the quantifier $\exists P$ over disjunction, it is enough to eliminate $\exists P$ in the following three formulas:

(1) $\exists P. \exists y. P(y)$
$\models\!\!\mid$ $\exists y. \exists P. \forall x. \big(x \not\approx y \vee P(x)\big) \wedge \big(\mathtt{true} \vee \neg P(x)\big)$
$\models\!\!\mid$ $\exists y \forall x. x \not\approx y \vee \mathtt{true}$
$\models\!\!\mid$ $\mathtt{true}$

(2) $\exists P. \exists y. \neg P(y)$
$\models\!\!\mid$ $\exists y. \exists P. \forall x. \big(x \not\approx y \vee \neg P(x)\big) \wedge \big(\mathtt{true} \vee P(x)\big)$
$\models\!\!\mid$ $\exists y \forall x. x \not\approx y \vee \mathtt{true}$
$\models\!\!\mid$ $\mathtt{true}$

(3) $\exists P.\,\big(\exists y.\,P(x)\big) \wedge \big(\exists y.\,\neg P(x)\big)$
$\models\!\!\models \quad \exists y_1 y_2.\,\exists P.\,\big(\forall x.\,x \not\approx y_1 \vee P(x)\big) \wedge \big(\forall x.\,x \not\approx y_2 \vee \neg P(x)\big)$
$\models\!\!\models \quad \exists y_1 y_2 \forall x.\,x \not\approx y_1 \vee x \not\approx y_2$
$\models\!\!\models \quad \exists y_1 y_2.\,y_1 \not\approx y_2$

Hence, φ is semantically equivalent to

$$\begin{aligned}&\big(\forall x_1 x_2 x_3.\,R(x_1,x_2) \vee \neg R(x_1,x_3)\big)\\ &\wedge \Big(\big(\forall x_1 x_2.\,R(x_1,x_2)\big) \vee \big(\forall x_1 x_2.\,\neg R(x_1,x_2)\big) \vee \big(\exists y_1 y_2.\,y_1 \not\approx y_2\big)\Big)\ .\end{aligned}$$

Remark 7.3.7. *Several remarks regrading the shape of the resulting formulas in Theorem 7.3.5 are in order. (a) Although the elimination of $\exists P$ potentially introduces new (dis)equations, these only involve existentially quantified and free variables. This means, the separation conditions are not violated by these newly introduced equations. Hence, the introduction of such atoms in one elimination step does not pose an obstacle to the iterated elimination of multiple existential second-order quantifiers. (b) As the subformulas $\chi_1^*(v_{i_2}, \bar{\mathrm{z}})$ may contain universal quantifiers $\forall w$ and atoms $R(\ldots w \ldots v_{i_2} \ldots)$, the separateness condition regarding universally and existentially quantified variables might be violated when introducing the subformulas $\chi_1^*(v_{i_2}, \bar{\mathrm{z}})$ and, similarly, the subformulas $\chi_2^*(u_{i_1}, \bar{\mathrm{z}})$. (c) Perhaps more severely, the introduction of atoms $R(\ldots w \ldots v_{i_2} \ldots)$ may create co-occurrences of variables from the sets $\widetilde{\mathrm{x}}_k$ and $\widetilde{\mathrm{y}}_i$, if $w \in \widetilde{\mathrm{x}}_k$ and $v_{i_2} \in \widetilde{\mathrm{y}}_i$. Then, the sets $\widetilde{\mathrm{x}}_k$ and $\widetilde{\mathrm{y}}_i$ are not separated anymore in formulas that contain the new atom. Similar effects might affect pairs $\widetilde{\mathrm{x}}_k, \widetilde{\mathrm{x}}_{k'}$ and $\widetilde{\mathrm{y}}_i, \widetilde{\mathrm{y}}_{i'}$. Hence, if we were to predict whether elimination of the two second-order quantifiers in a formula $\exists Q \exists P.\,\varphi$ is possible using the methods outlined above, we would need to predict which sets of variables will be separated after the elimination of $\exists P$.*

The above observations seem to indicate that it is not straightforward to formulate a version of Theorem 7.3.5 that neatly facilitates iterative elimination of multiple quantifiers in second-order SF. On the other hand, it might be worthwhile to base the theorem on a second-order variant of GBSR instead, as Observation (b) might cause fewer troubles in the GBSR setting. Another interesting aspect is that the symmetry regarding the two Conditions (1) and (2) in Theorem 7.3.5 is perhaps more restrictive than necessary. It seems that Condition (2) is obsolete, as the resulting formula in Lemma 7.3.3 could be generalized in such a way that the restriction imposed by (2) is not satisfied but second-order quantifiers can still be eliminated. Altogether, it is subject to future investigations whether Theorem 7.3.5 can be enhanced to facilitate iterative elimination of multiple quantifiers.

The presented result can only be a first step towards the formulation of a novel fragment of second-order logic that (i) extends the monadic second-order fragment, (ii) is based on the concept of separateness, (iii) admits elimination of second-order quantifiers, also in an iterated fashion. Remark 7.3.7 already makes clear that a lot remains to be done, in order to achieve this goal. Furthermore, there seems to be no good reason to confine ourselves to the elimination of quantifiers over unary predicates only, but aim for higher arities as well. Moreover, the requirements regarding separateness of variables could be weakened by taking boolean structure into account instead of only concentrating on the atoms in a given formula, compare also Section 3.6. For example, the formula $\exists P.\,\forall xy.\,P(x) \wedge \big(P(y) \vee R(x,y)\big)$ does not satisfy the prerequisites of Theorem 7.3.5, as $\{x\}$ and $\{y\}$ are not separated and the set $\{x,y\}$ contains two variables that occur as arguments of P. However, the theorem can be applied to the equivalent formula $\exists P.\,\forall x_1 x_2 y.\,P(x_1) \wedge \big(P(y) \vee R(x_2,y)\big)$, as the sets $\{x_1\}$ and $\{x_2, y\}$ are separated and x_2 does not occur as argument of P. As a third possible improvement, equations between universal and existential variables should be allowed in a less restrictive way than they are at the moment. To this end, some of the methods that are used to handle equations during quantifier elimination in the monadic second-order fragment might be applicable in the more general setting as well.

In the outlined approach we concentrated on transforming the input formulas syntactically until the Basic Elimination Lemma (Proposition 7.3.4) is applicable. In future work, it is of course advisable to also try other known approaches, such as the ones described in [GSS08], e.g. the SCAN algorithm, the DLS* algorithm, hierarchical theorem proving, or variations thereof. The unmodifed

DLS algorithm, as presented in [GSS08], fails on the logic fragment described in Theorem 7.3.5. In particular, the preprocessing phase is not always able to transform the input into the required form, although this would be possible in principle. This is already true for monadic sentences such as $\varphi := \exists P. \forall x \exists y. \big(\neg P(x) \vee P(y)\big) \wedge \big(P(x) \vee \neg P(y)\big)$, which is equivalent to $\exists P. \forall x \exists y. P(x) \leftrightarrow P(y)$. Conradie gave a necessary and sufficient condition regarding the syntax of formulas in which DLS can successfully eliminate an existential second-order quantifier [Con06]. It turns out that the occurrences of P in φ violate Conradie's condition in many ways — every occurrence of P is in *malignant* conjunctions and disjunctions and inside a $\forall\exists$-*scope*. Nonetheless, it is not hard to see that there is a first-order formula that is equivalent to φ, namely `true`. A slight modification of the DLS preprocessing step in the spirit of Lemma 3.2.4 might already solve this particular issue.

Part II

First-Order Linear Arithmetic with Uninterpreted Predicates

Chapter 8

Linear Arithmetic with Uninterpreted Predicates

In Part I of the present thesis we have mainly treated first-order logic where only the distinguished equality predicate had a fixed semantics. The semantics of all other predicate symbols and function symbols was open to interpretation and only determined by the structures that we considered. In the literature, such symbols are sometimes referred to as *uninterpreted*, meaning that their interpretation is not fixed a priori. In Part II we will be interested in first-order languages where the vocabulary contains uninterpreted predicate and function symbols alongside interpreted symbols. Moreover, parts of the domain are fixed and other parts are not predetermined, i.e. we technically have a sorted setting. In particular, we shall be considering fragments of the language of linear arithmetic over the rationals or integers with additional uninterpreted predicate symbols and ask whether the associated satisfiability problems are decidable.

We have already encountered two well-known decidable arithmetic theories[1] in Chapter 2 (pages 20–22) and in Section 7.1: *Presburger arithmetic* — the first-order theory of the integers[2] with addition and the usual ordering — and *linear rational arithmetic* — the first-order theory of linear sentences over the rational numbers with the usual ordering. Both fragments admit quantifier elimination, first elimination procedures were devised by Presburger [Pre29] (see [Sta84] for an English translation and [End01] for a textbook exposition) and Tarski [Tar57], respectively. In 1974 the computational time complexity of deciding membership in the theory of Presburger arithmetic was shown to be doubly exponential by Fischer and Rabin [FR74], and a more precise lower bound was later derived by Berman [Ber80]. The computational complexity of deciding validity in linear rational arithmetic is at least exponential, as follows from results in [FR74], see also Section 7.4 in [BM07]. Weispfenning has shown that quantifier elimination in linear rational arithmetic requires at least doubly exponential time [Wei88]. A very recent account of the history and the current state of affairs regarding Presburger arithmetic can be found in [Haa18]. For a survey of the recent developments regarding quantifier elimination in *real closed fields* and some historical background see [Stu17, Stu18]; these methods also work for rational arithmetic formulas as long as they are linear. Both arithmetic languages found numerous applications. For a textbook introduction to linear arithmetic over the rationals or over the integers in the context of automated reasoning and verification see, e.g., Chapters 7 and 8 in [BM07] and Chapter 5 in [KS16].

Undecidability of First-Order Arithmetic with Uninterpreted Predicate Symbols

It has been discovered more than half a century ago that the addition of a single uninterpreted unary predicate symbol to Presburger arithmetic renders the associated satisfiability problem (and

[1] Formally, a logical Σ-theory $\mathcal{T}$ is considered *decidable*, if there is an algorithm that can decide membership of any given Σ-sentence in $\mathcal{T}$.

[2] Originally, Presburger arithmetic considers the domain of the nonnegative integers. But this is not serious restrictions, see, e.g., Example 3.7 in [BM07].

also the associated validity problem) undecidable. In 1957 Putnam [Put57] discussed this theory as one example of an undecidable first-order theory that is somewhat stronger than the decidable first-order theory of natural numbers with the successor function and a single uninterpreted unary predicate symbol.[3] Lifshits mentioned in a note [Lif69] (without giving a proof) that the addition of one uninterpreted predicate symbol — of unspecified arity — to Presburger arithmetic leads to undecidability. In the technical report [Dow72] Downey gave an encoding of two-counter machines[4] and their halting problem in the Horn fragment of Presburger arithmetic with a single unary predicate symbol that is uninterpreted. Moreover, undecidability is also implied by a general result due to Garfunkel and Schmerl [GS74] published in 1974. Seventeen years later Halpern [Hal91] strengthened the undecidability result for Presburger arithmetic with a single uninterpreted unary predicate symbol in that he proved Π^1_1-completeness of the associated validity problem, which, given that the considered language is closed under negation, entails Σ^1_1-completeness of satisfiability.[5] Only recently, Speranski [Spe13b] gave an alternative characterization of the analytical hierarchy that is based on a reduction of Π^1_n-formulas with multiplication to Π^1_n-formulas without multiplication but with an uninterpreted unary predicate symbol. Halpern's Π^1_1-completeness can be conceived as a special case of this more general point of view. We shall add some results to this line of contributions in Chapter 11. As a starter, we will give simple encodings of the halting problem for two-counter machines based on $\exists^*\forall^*$ sentences with a very restricted arithmetic language and a single uninterpreted predicate symbol whose arity is greater than one. On the arithmetic side we get along with one of the following fragments where $c \in \mathbb{Q}$ and $\lhd$ ranges over the relations $<, \leq, =, \neq, \geq, >$: *difference constraints* $x - y \lhd c$, *additive constraints* $x + y \lhd c$, *quotient constraints* $x \lhd c \cdot y$, and *multiplicative constraints* $x \cdot y \lhd c$. The details of the encodings can be found in Section 11.1. In the rest of Chapter 11, we will focus on the universal fragment of Presburger arithmetic plus a single unary uninterpreted predicate symbol. We shall devise a novel encoding of two-counter machines and investigate several variants of it. Our results will shed more light on the border between decidability and undecidability in this context. We will also see that allowing for a $\forall\exists$ quantifier alternation even leads to Σ^1_1-completeness (of satisfiability). Furthermore, we shall assess the relevance of the undecidability results for a number of verification frameworks. The mentioned results will not change substantially when we use the rationals as underlying domain instead of the integers.

Decidable First-Order Arithmetic Fragments with Uninterpreted Predicate Symbols

All of the above said leads to one conclusion: In order to obtain decidable subfragments of the combination of linear arithmetic with uninterpreted predicate symbols, the arithmetic part needs to be restricted considerably.[6] We shall explore the decidable side in Chapter 10 and investigate two decidable fragments. Both will be an extension of the Bernays–Schönfinkel–Ramsey fragment (BSR) with a restricted form of linear-arithmetic constraints. For notational convenience, we use the notation $\Lambda \wedge \Gamma \to \Delta$ for BSR clauses with arithmetic constraints, where Λ and Γ are conjunctions of atoms, respectively, and Δ is a disjunction of atoms. The part Λ contains exclusively arithmetic atoms and no uninterpreted symbols. The parts Γ and Δ, on the other hand, only contain atoms that are either (a) relational atoms with some uninterpreted predicate symbol or (b) non-arithmetic equations $u \approx v$, where u and v are implicitly universally quantified first-order variables of a sort that is not pre-determined, i.e. of an uninterpreted sort. We extend BSR in two ways with linear-arithmetic expressions and call the obtained clause fragments *BSR with simple linear rational*

[3] Büchi [Büc60, Büc62] and Rabin [Rab69] proved that the theory remains decidable, if an arbitrary number of uninterpreted unary predicate symbols is admitted.

[4] Two-counter machines are a special case of Minsky machines, see [Min67], Sections 11 and 14. See also Section 11.1 of the present thesis.

[5] For the definition of the analytical hierarchy and the sets Π^1_1 and Σ^1_1, see, e.g., Chapter IV.2 in [Odi92] or Chapter 16 in [Rog87].

[6] There are still alternatives. One possibility could be to consider sentences in CNF and restrict the occurrences of variables as arguments of uninterpreted predicate symbols in clauses, as is done in certain decidable clause classes (cf. Chapter 3, pages 27–28). This was pointed out by Christoph Weidenbach in a discussion in January 2019. We shall not consider this approach any further in the present thesis.

constraints — BSR(SLR) — and *BSR with bounded difference constraints — BSR(BD).*

In the first clause class, which we shall treat in detail in Section 10.2, we allow arithmetic atoms of the form $s \lhd t$, $x \lhd t$, and $x \lhd y$ in the Λ-part of clauses, where x and y are rational-valued variables that are implicitly universally quantified, s and t are linear arithmetic terms that are variable free (ground), and $\lhd$ ranges over $<, \leq, =, \neq, \geq, >$. The ground terms s and t may contain uninterpreted constant symbols of sort $\mathbb{Q}$. Since their value is not predetermined, they can be conceived as being existentially quantified. An exemplary BSR(SLR) clause is BSR(SLR)

$$d > 0 \wedge c - 1 \leq x \wedge x < c + 2d \wedge y < x \wedge Q(y) \rightarrow P(x, y).$$

Remark 8.0.1. *The arithmetic atoms admitted in BSR(SLR) are similar to the kind of arithmetic atoms that appear in the context of the* array property fragment *[BMS06, Bra07] and extensions thereof (see, e.g., [GdM09, HVW17a]).*[7] *The array property fragment can be used in the context of verification to specify certain properties of array data structures (see [KS16], Chapter 7 or [BM07], Chapter 11 for an introduction and examples). Apart from the fact that we concentrate on the rational domain instead of the integers in Section 10.2, the main difference is that we allow strict inequalities and disequations between universally quantified variables. In the presence of uninterpreted non-constant function symbols, strict inequality or disequations can be used to assert that some uninterpreted function f is injective. This expressiveness prevents certain instantiation-based approaches to satisfiability checking from being applicable, e.g. the methods in [BMS06, Bra07]. In the context of the array property fragment, this expressiveness even leads to undecidability, see, e.g. Section 2.4 in [Bra07], or Theorem 11.16 in [BM07].*

A close relative of the array property fragment is the hashtable property fragment *presented in [Bra07], see also [BM07], Section 11.3. It admits the same syntax for arithmetic atoms.*

In the BSR(BD) clause class, treated in Section 10.4, we allow arithmetic atoms of the form $x \lhd c$, $x \lhd y$, and $x - y \lhd c$ in the Λ-part of clauses, where x and y are rational-valued variables, c could be any rational number, and $\lhd$ ranges over $<, \leq, =, \neq, \geq, >$ again. We refer to atoms of the form $x - y \lhd c$ as *difference constraints*. An exemplary BSR(BD) clause is BSR(BD)

$$x - y < 1 \wedge -2 \leq x \wedge x \leq 2 \wedge -1 \leq y \wedge y \leq 3 \wedge y < z \wedge Q(y, z) \rightarrow P(x, y) \vee P(y, x).$$

Remark 8.0.2. *Already in the seventies, Pratt identified difference constraints and Boolean combinations thereof as an important tool for the formalization of verification conditions [Pra77].*[8] *Applications include the verification of timed systems and scheduling problems (see, e.g. [Pra77, NMA+02, TSSP04, dMB11], the textbook [KS16] (Section 5.7), the handbook article [BT18] (Section 11.4.5), and Mahfoudh's PhD thesis [Mah03] for references). Dedicated decision procedures for Boolean combinations of difference constraints have been devised, see, e.g. [SSB02, MNAM02, CAMN04, ACGM04, NO05, CM06, WGG06] and the references therein.*

As unrestricted combinations of uninterpreted predicate symbols with difference constraints lead to an undecidable satisfiability problem (once more, two-counter machines and their halting problem can be encoded, see Sections 11.1 and 11.4), we have to further confine the language. We require that every difference constraint $x - y \lhd c$ has to be conjoined with four additional constraints $c_x \leq x$, $x \leq d_x$, $c_y \leq y$, $y \leq d_y$, where c_x, d_x, c_y, d_y are rationals. This restriction seems to weaken expressiveness severely. Indeed, it has to, since we aim for a decidable satisfiability problem. Yet, we show in Section 10.5 that BSR(BD) clause sets are expressive enough to formulate the reachability problem for timed automata, for instance.

The main result of Chapter 10 is that the satisfiability problems associated with BSR(SLR) and BSR(BD) are decidable (cf. Theorems 10.2.14 and 10.4.10). Both results have a very similar proof outline, which, roughly speaking, proceeds as follows. Given some satisfiable sentence $\varphi := \exists \bar{z} \forall \bar{x}.\ \bigwedge_i C_i(\bar{z}, \bar{x})$, where the C_i are clauses adhering to the syntactic restrictions of BSR(SLR) or BSR(BD), and given any model $\mathcal{A} \models \varphi$, we define a suitable equivalence relation $\sim$ over tuples of rationals such that $\sim$-equivalent tuples cannot be distinguished by the arithmetic constraints

[7] This was brought to the attention of the author of the present theses by Viorica Sofronie-Stokkermans at the VTSA summer school in Koblenz, Germany, in August 2015

[8] In parts of the literature difference constraints are referred to as *separation predicates* [SSB02, TSSP04].

occurring in φ. We can formulate this property more precisely as follows. Let $\bar{\mathrm{x}}'$ be the restriction of $\bar{\mathrm{x}}$ to rational-valued variables. Moreover, let $\bar{\mathsf{a}}$ be a tuple of elements from $\mathcal{A}$'s domain such that $\mathcal{A} \models \forall \bar{\mathrm{x}}.\, \varphi(\bar{\mathsf{a}}, \bar{\mathrm{x}})$ — notice that apart from rationals $\bar{\mathsf{a}}$ may also contain elements from some uninterpreted sort. For any two $\sim$-equivalent tuples $\bar{r}, \bar{s} \in \mathbb{Q}^{|\bar{\mathrm{x}}'|}$ we require for all conjunctions of constraints $\Lambda(\bar{\mathrm{z}}, \bar{\mathrm{x}}')$ in φ that $\mathbb{Q} \models \Lambda(\bar{\mathsf{a}}, \bar{r})$ if and only if $\mathbb{Q} \models \Lambda(\bar{\mathsf{a}}, \bar{s})$. The key to decidability is that we choose $\sim$ so that it induces only finitely many equivalence classes. Based on $\mathcal{A}$, $\bar{\mathsf{a}}$, and $\sim$, we then construct a model $\mathcal{B} \models \varphi$ that interprets the uninterpreted sorts with finite sets and whose interpretations of uninterpreted predicate symbols do not distinguish $\sim$-equivalent tuples of rational numbers. We call structures of this form *uniform* with respect to $\sim$. The proof of $\mathcal{B}$'s existence is partially based on basic methods from Ramsey theory. If $\mathcal{A}$ contains a certain collection $\overline{Q}$ of finite sets $Q_1, \ldots, Q_k \subseteq \mathbb{Q}$ such that any two $\sim$-equivalent tuples over elements from $\overline{Q}$ are indistinguishable under $\langle \mathcal{A}, \bar{\mathsf{a}} \rangle$, then $\mathcal{B}$ treats any tuple $\bar{r}$ over $\mathbb{Q}$ like the $\sim$-equivalent tuples $\bar{q}$ over $\overline{Q}$. For this approach to work, it is essential that $\overline{Q}$ covers all $\sim$-equivalence classes over $\mathbb{Q}$ and each Q_i contains a critical mass of rational numbers. Ramsey theory provides the right methods to show that such sets exist for any pair $\langle \mathcal{A}, \bar{\mathsf{a}} \rangle$ with $\mathcal{A} \models \forall \bar{\mathrm{x}}.\, \bigwedge_i C_i(\bar{\mathsf{a}}, \bar{\mathrm{x}})$. The guaranteed existence of models that are uniform with respect to an equivalence relation $\sim$ inducing only a finite number of equivalence classes is similar to the *finite model property* in entirely uninterpreted settings and it immediately implies decidability.

Example 8.0.3. *Consider the following BSR(SLR) sentence φ_1 and the BSR(BD) sentence φ_2 (we use convenient notation that could easily be converted into syntax that adheres to the restrictions posed by BSR(SLR) and BSR(BD)):*

$$\begin{aligned}\varphi_1 := \exists z_1 z_2 \forall xy.\, \big(3 < z_1 < z_2 < \tfrac{31}{5}\big) \wedge \big(3 \leq x < z_1 \wedge\ 3 < y < z_2 \ \rightarrow\ P(x,y)\big)\\ \wedge \big(z_1 \leq x < y \wedge\ \ y < \tfrac{31}{5} \ \rightarrow\ P(x,y)\big)\end{aligned}$$

and

$$\begin{aligned}\varphi_2 := \forall xy.\, \Big(-2 < x - y < 0 \wedge\ -2 \leq x \leq 2 \wedge\ -2 \leq y \leq 2 \ \rightarrow\ P(x,y)\Big)\\ \wedge \Big(\ \ x \leq -2 \wedge\ \ -2 < y < 0 \ \rightarrow\ P(x,y)\Big)\\ \wedge \Big(\ \ 1 \leq x \wedge\ \ 1 < y < 2 \ \rightarrow\ P(x,y)\Big)\ .\end{aligned}$$

Figure 8.1 illustrates suitable equivalence relations $\sim_1$ and $\sim_2$ for dimension two (we only have two universally quantified variables per sentence and per clause). Moreover, satisfying uniform interpretations are depicted for the predicate symbol P in φ_1 and φ_2, respectively. Decidability of the associated satisfiability problem follows from the fact that $\sim_1$ and $\sim_2$ induce only finitely many equivalence classes.

The outlined approach to proving decidability is quite general. For a given language combining linear rational arithmetic with uninterpreted predicate symbols we just have to find a suitable equivalence relation $\sim$ and the rest can be developed along the same lines as outlined above. Hence, the outlined approach may turn out to be applicable to other fragments as well. Moreover, in the light of the insights we have gained in Chapter 3 it is possible to generalize the decidability results for BSR(SLR) and BSR(BD) to SF and GBSR variants of the two fragments, cf. Theorem 10.3.2 and Corollary 10.3.4.

More on Related Work

We have already elaborated on undecidable fragments of first-order arithmetic with uninterpreted predicate symbols. There is also a number of works describing decidable fragments of first-order logic in which linear rational or integer arithmetic is mixed with uninterpreted function or predicate symbols. However, the results seem to be scattered across the literature. We shall report on a few such results in what follows. An early result can be found in [Put57]: the first-order theory of the natural numbers with the successor function plus a single uninterpreted unary predicate symbol is decidable. This result is subsumed by results due to Büchi [Büc60, Büc62], later extended by Rabin [Rab69], who show that the monadic second-order theory of the natural numbers with the successor function (today also known as the *monadic second-order theory of one successor (S1S)*)

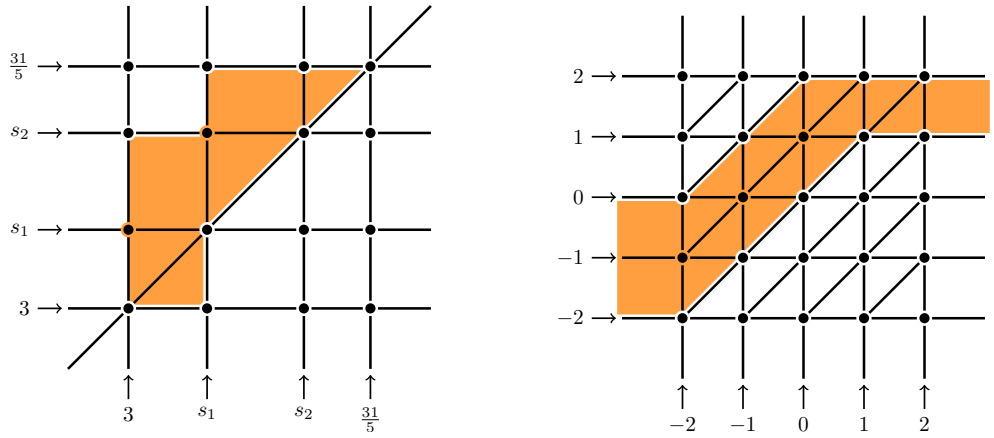


Figure 8.1: Left: Partition of the two-dimensional rational plane into $\sim_1$-equivalence classes with respect to two rational values s_1, s_2 assigned to z_1, z_2 in φ_1, respectively. Right: Partition of the two-dimensional rational plane into $\sim_2$-equivalence classes with respect to the sentence φ_2. Every dot, line segment, rectangular white area, and triangular white area represents an equivalence class. Moreover, the orange-colored parts represent subsets of the rational plane that represent interpretations of the predicate $P(x, y)$ that are satisfying for φ_1 and φ_2, respectively, if we put the x-axis horizontally and the y-axis vertically. These interpretations are uniform in the sense that any of the equivalence classes is either completely contained in the respective subset or it is disjoint from the subset.

is decidable. Shelah [She75] reproved some of these results using different methods; the article also contains a historical overview and additional references to related works. Moreover, Shelah showed in the same article that the monadic second-order theory of the usual order over the real numbers is undecidable. Ferrante and Rackoff [FR79] investigated the computational complexity for deciding the monadic first-order theory of one successor with an uninterpreted predicate symbol: they obtain doubly exponential upper and lower bounds.

The satisfiability problem for existential first-order sentences combining linear rational or integer arithmetic with uninterpreted function and predicate symbols can be shown to be decidable using the *Nelson–Oppen combination framework* (see Section 10.3). Pratt [Pra77] elaborated on the subcase where the arithmetic atoms are restricted to *difference constraints*, i.e. atoms of the form $x - y \triangleleft c$ where c ranges over the integers. This kind of arithmetic atoms will play a major role in Sections 10.4 and 11.4. On the one hand, we show that satisfiability is decidable for a certain first-order fragment combining *bounded* difference constraints with uninterpreted predicate symbols of arbitrary arity in the former section. In the latter section, on the other hand, we show how the halting problem of two-counter machines can be encoded using only arithmetic atoms of the form $x - y \triangleleft c$ where c is an uninterpreted arithmetic constant.

An interesting subcase of the existential first-order fragment of linear integer arithmetic with uninterpreted function symbols with a decidable satisfiability problem is *counter logic*. This fragment was motivated and investigated in [BLS02] in the context of hardware verification. The arithmetic part is restricted to the positive integers with the successor function and the predecessor function (see Section 11.5.3 for more details). Suitable decision procedures were presented in [BLS02, GHN$^+$04, ABRS09]. In [ABRS09] also the case of successor and predecessor modulo some fixed integer is treated. Further positive results stem from the field of software verification, where data structures, such as arrays, or memory are often formalized using uninterpreted function symbols and restricted forms of arithmetic over integer indices, partly allowing for universal quantification over indices. See, for example, [BMS06, HIV08, ABRS09, GdM09, KPSW10, Sof14, RIS17, HVW17a] and the references therein. See also the textbooks [BM07, KS16] and the handbook article [BT18]

for further references. We will show in Section 11.5 that little extensions of several of the mentioned fragments lead to undecidable satisfiability problems.

There are also decidability results based on theory combination beyond the Nelson–Oppen case, involving universal quantification. Interesting results in this direction are due to Fontaine and his collaborators [Fon07, Fon09, AF11, CFR14], where component theories are considered that can be expressed in the decidable first-order fragments $\mathrm{MFO}_{\approx}$, BSR, AF with equality, GF, LGF, or FO^2 (cf. Sections 10.3 and 12.1.1).

Finally, there is the vast field of *constraint logic programming* (see, for instance, [FA03] for an introduction), where one can also find positive results regarding first-order arithmetic with uninterpreted predicate symbols, e.g. [CMT92, CM93].

To the best knowledge of the author, the decidability results that come closest to the results developed in Chapter 10 of the present thesis can be found in [GdM09, KW12, Kru13, FW12, Fie13]. In [GdM09] Ge and de Moura presented a very general instantiation approach that is a decision procedure for certain subfragments of linear integer and rational arithmetic with uninterpreted function and predicate symbols. For instance, BSR(SLR) without strict inequalities and without arithmetic disequations can be decided using this instantiation method. It fails, however, in the presence of $<, \neq, >$ in arithmetic atoms in BSR(SLR) clauses. In [KW12, Kru13] Kruglov and Weidenbach devise a procedure based on *hierarchic superposition* that decides satisfiability for the existential first-order fragment of arithmetic with uninterpreted predicate symbols. The calculus is not limited to the language of arithmetic though, but also serves as a decision procedure for other background theories. It can even handle universal quantification in Horn clause sets as long as all variables in background-theory terms are existentially quantified. The setting of linear and nonlinear arithmetic with uninterpreted predicate symbols was treated in [AKW09, EKK$^+$11] in particular. In [FW12, Fie13] Fietzke and Weidenbach described an encoding of the *reachability problem* for *timed automata* in a fragment of first-order linear arithmetic with uninterpreted predicate symbols whose arity depends on the number of clocks in the input automaton (cf. Section 10.5). Then, they show that Kurglov and Weidenbach's *hierarchic superposition calculus modulo linear arithmetic* [AKW09, KW12, FW12] can decide (un)satisfiability for the resulting logic fragment. As we have already mentioned above, we prove in Section 10.5 that the reachability problem for timed automata can be formalized using BSR(BD) clause sets. However, the latter does not cover all the clauses that result from the encoding of Fietzke and Weidenbach, called the *FOL(LA) encoding* of timed automata in Section 10.5 (cf. Definition 10.5.7). The two fragments are syntactically incomparable: while the FOL(LA) encoding of timed automata results in Horn clause sets and contains essentially arithmetic atoms of the form $u - v = x - y$, BSR(BD) is not restricted to Horn clauses and does not admit arithmetic atoms $u - v = x - y$. We shall show in Section 10.5 that this form of atoms is not required to capture the reachability problem for timed automata.

One direction to look for new decidable fragments is to draw inspiration from similar encodings of extensions of the timed automaton formalism. There are numerous kinds of extensions of timed automata that have been proposed and might be worthwhile targets for such an approach, see, e.g. the handbook articles [BFL$^+$18, DFPP18] for references. Another source of inspiration might come from *metric temporal logic (MTL)*, a family of extensions of *linear-time temporal logic (LTL)*. In MTL the usual temporal operators of LTL, e.g. *eventually* and *until*, are enhanced in such a way that certain quantitative timing constraint can be expressed. An overview and more references can be found in [OW08, HOD17, BLM$^+$17], for example.

Chapter 9

Additional Technical Preliminaries

We take over and reuse the basic notions and notation introduced in Part I, in particular in Chapter 1. In Part II we mainly consider many-sorted first-order logic with equality and a mixture of interpreted and uninterpreted predicate and function symbols. Some of the terminology we shall use is borrowed from the framework of *hierarchic combinations of uninterpreted first-order logic with background theories* due to Bachmair, Ganzinger, and Waldmann, see [BGW94, BW13b, BW13a]. In particular, in order to simplify terminology and definitions, we take over the neat distinction of the interpreted part of the considered language, ambiguously referred to as *background theory* or *base theory* (together with *background/base sorts*), from the *uninterpreted part*. The latter comprises uninterpreted sorts and uninterpreted predicate and function symbols, all of which have to be assigned a meaning in terms of structures. Although this is technically not necessary, we mostly restrict our attention to single-sorted background theories, such as linear rational or linear integer arithmetic with the base sorts $\mathbb{Q}$ and $\mathbb{Z}$, respectively.[1] Similarly, we most of the time only consider a single uninterpreted sort $\mathcal{S}$ (we sometimes also use the term *free sort*). This sort needs to be interpreted with some nonempty domain, as usual. We continue to use the symbol $\approx$ to denote the built-in equality predicate for sort $\mathcal{S}$ — for the arithmetic sort we use the sign $=$ to denote the identity relation. $\approx$, $=$

In accordance with this division, we use pairwise-disjoint, countably infinite sets of variables $\mathrm{Var}_{\mathbb{Q}}, \mathrm{Var}_{\mathbb{Z}}, \mathrm{Var}_{\mathcal{S}}$ of the respective sorts. Moreover, vocabularies now come equipped with sort information. An uninterpreted predicate symbol P, for instance, can have a mixed-sort signature $P : \xi_1 \times \ldots \times \xi_m$, where the ξ_i can be any of the sorts $\mathbb{Q}, \mathbb{Z}, \mathcal{S}$. To avoid confusion, we tacitly assume that no predicate or function symbol is overloaded, i.e. each of them has a unique sort.

Recall the definition of linear rational arithmetic (LRA) terms and formulas from Section 7.1. The underlying vocabulary is $\Sigma_{\mathrm{LRA}} := \langle\{<, \leq, =, \neq, \geq, >\}, \mathbb{Q} \cup \{+, \cdot\}\rangle$, where $\mathbb{Q}$ is the only occurring sort. *LRA terms* are all Σ_{LRA}-terms in which multiplication only occurs in (sub)terms of the form $r \cdot x$ where r is a rational coefficient and x is a first-order variable of sort $\mathbb{Q}$. Variables of other sorts are not admitted in LRA terms. For convenience, we use abbreviations such as $-\frac{1}{2}x - y$ for the formal expression $-\frac{1}{2} \cdot x + (-1) \cdot y$ and the like. *LRA formulas* are all first-order Σ_{LRA}-formulas in which all terms are LRA terms. The terms of Presburger arithmetic, called *PA terms*, are defined to be the terms over the vocabulary $\Sigma_{\mathrm{PA}} := \langle\{<, \leq, =, \neq, \geq, >\}, \{0, 1, +, -\}\rangle$, where the only sort is $\mathbb{Z}$. We also use convenient abbreviations for PA terms, such as $-3x + 2y$ for the formal expression $0 - (x + x + x) + y + y$.[2] In Chapter 11 we shall consider the extended language of PA+P *terms* and PA+P *formulas* which is based on the vocabulary $\Sigma_{\mathrm{PA}+P} := \langle\{<, \leq, =, \neq, P\}, \{0, 1, +, -\}\rangle$ Σ_{LRA} *LRA terms* *PA terms* Σ_{PA} PA+P *terms* $\Sigma_{\mathrm{PA}+P}$

[1] We use symbols such as $\mathbb{Q}$, $\mathbb{Z}$, $\mathbb{N}$, and $\mathbb{R}$ with different meaning. Depending on the current context, we use them to address the respective sets of numbers, structures, or sorts.

[2] Notice that the formal term length of the LRA term $t_1 := 4x - 2$ differs significantly from the length of the PA term $t_2 := 4x - 2$. While the former has term length $\mathrm{len}(t_1) = \mathrm{len}(4 \cdot x + (-2)) = 5$, the latter has length $\mathrm{len}(t_2) = \mathrm{len}(x+x+x+x-1-1) = 11$. More generally, the rational numbers in LRA terms are formally represented by constant symbols, the integers in (abbreviated) PA terms can be conceived as being represented in a unary encoding. Since we will not be concerned with PA terms when analyzing computational complexity of certain decision procedures later, this technical curiosity will not impact our investigations any further.

where P is an uninterpreted unary predicate symbol of sort $\mathbb{Z}$. Furthermore, we shall also consider a second extended language: LRA+PN *terms* and LRA+PN *formulas*. Both are based on the vocabulary $\Sigma_{\mathrm{LRA}+PN} := \langle\{<, \leq, =, \neq, P, N\}, \{0, 1, +, -\}\rangle$ with P, N being uninterpreted unary predicate symbols of sort $\mathbb{Q}$.

LRA+PN *terms*, $\Sigma_{\mathrm{LRA}+PN}$

Throughout the following chapters we shall concentrate on clauses (disjunctions of literals) and sets of clauses. Occasionally, we treat sets N of clauses as if they were sentences. Then, we consider any variables that occur in N as implicitly universally quantified. More precisely, given any clause set N that is finite, it can be conceived as a sentence $\varphi_N := \forall \bar{\mathrm{x}}.\ \bigwedge_{C(\bar{\mathrm{x}}) \in N} C(\bar{\mathrm{x}})$, where $\bar{\mathrm{x}}$ is a tuple collecting all variables occurring in N. Existentially quantified variables are represented by uninterpreted constant symbols, called *Skolem constants*, of the respective sort, but their value is not predetermined. This means, we implicitly restrict our attention to the $\exists^*\forall^*$ prefix class, if not explicitly stated otherwise. Given any clause C, we use the following notation: the set of all constant symbols occurring in C is denoted by $\mathrm{consts}(C)$ — this includes rational numbers or integers. Similar notation is used for other syntactic objects, e.g. clause sets.

$\mathrm{consts}(C)$

For finite clause sets N we define $\mathrm{len}(N) := \sum_{C \in N} \mathrm{len}(C)$ and $\|N\| := \sum_{C \in N} \|C\|$. Since the vocabulary underlying any formula φ cannot be assumed to be finite if φ contains LRA terms, the encoding length $\|\varphi\|$ of such formulas can in general not be expressed in terms of $\mathrm{len}(\varphi)$ alone. Instead, the bit length of the rational numbers occurring in φ has to be taken into account as well. Therefore, we redefine our notion of *encoding length* accordingly. Henceforth, we assume that $\|\varphi\| \in \mathcal{O}\big(\mathrm{len}(\varphi) \cdot \log(|\Pi| + |\Omega| + |\mathrm{vars}| + \kappa)\big)$ where Π and Ω are the sets of predicate and function symbols occurring in φ and κ is the smallest integer that is larger than the absolute value of any numerator and denominator occurring in any rational number in φ (represented by an equivalent *irreducible fraction*). The same applies to clause sets containing LRA terms.

$\mathrm{len}(N)$, $\|N\|$

In the following chapters a *Bernays–Schönfinkel–Ramsey clause (BSR clause)* is understood to be a disjunction of literals that may contain constant symbols but no function symbols of positive arity. In order to denote BSR clauses that contain arithmetic constraints alongside with uninterpreted symbols, we use the notation $\Lambda \wedge \Gamma \to \Delta$, where Λ and Γ are conjunctions of atoms, respectively, and Δ is a disjunction of atoms. Since $\wedge$ and $\vee$ bind stronger than $\to$, explicitly putting the implicit parentheses yields $(\Lambda \wedge \Gamma) \to (\Delta)$. The part Λ contains exclusively arithmetic atoms and no uninterpreted symbols. The parts Γ and Δ, on the other hand, only contain atoms that are either (a) relational atoms with some uninterpreted predicate symbol or (b) non-arithmetic equations $u \approx v$, where u and v are first-order variables of a sort that is not pre-determined, i.e. of an uninterpreted sort. Requiring the parts Γ and Δ of clauses to not contain any arithmetic terms apart from variables does not limit expressiveness. First of all, for every implication $\Lambda \wedge \Gamma \to \Delta \vee s \lhd t$ where $\lhd \in \{<, \leq, =, \neq, \geq, >\}$ there is an equivalent implication $s \lhd' t \wedge \Lambda \wedge \Gamma \to \Delta$ where $\lhd' \in \{<, \leq, =, \neq, \geq, >\}$ is the negated counterpart of $\lhd$. Hence, every clause is equivalent to some clause in which all purely arithmetic atoms occur in the Λ part. Moreover, every arithmetic term t in Γ or Δ that is not a variable and that is not part of a purely arithmetic atom can be safely replaced with a fresh base-sort variable x_t when an arithmetic constraint $x_t = t$ is added to the Λ part of the clause (a process known as *purification* or *abstraction* [BGW94, KW12]).

BSR clause

$\Lambda \wedge \Gamma \to \Delta$

purification

Concerning semantics, we adhere to the definitions given in Chapter 1. However, we tacitly assume that all structures that we shall consider in the following chapters interpret the arithmetic function and predicate symbols in the usual way, unless explicitly stated otherwise. Notice that this also affects notions such as *semantic entailment* and *semantic equivalence*, which are implicitly re-defined based on the restriction to the mentioned class of structures. Given any clause set N, we occasionally write $\mathcal{A} \models N$ if we have $\mathcal{A} \models \varphi_N$ for the associated sentence $\varphi_N := \forall \bar{\mathrm{x}}.\ \bigwedge_{C(\bar{\mathrm{x}}) \in N} C(\bar{\mathrm{x}})$, where $\bar{\mathrm{x}}$ is a tuple collecting all variables occurring in N. This notation takes into account that all variables in N are implicitly considered to be universally quantified. Similarly, we extend the notions $\mathcal{A}$ *satisfies* N and $\mathcal{A}$ *is a model of* N to clause sets, if we actually mean "$\mathcal{A}$ satisfies φ_N" or "$\mathcal{A}$ is a model of φ_N" for the associated sentence φ_N. Furthermore, we tacitly assume that all considered variable assignments and substitutions respect sorts, i.e. given any first-order variable x of sort $\mathbb{Q}$ (or $\mathbb{Z}$), any variable assignment β, and any substitution σ, we assume that $\beta(x)$ is a value from $\mathbb{Q}$ (or $\mathbb{Z}$) and that the term $\sigma(x)$ has the sort $\mathbb{Q}$ (or $\mathbb{Z}$). We assume the same for any

$\mathcal{A} \models N$

first-order variable that is of any uninterpreted sort $\mathcal{S}$ (in the context of a given structure $\mathcal{A}$ that interprets $\mathcal{S}$ with some nonempty set $\mathcal{S}^{\mathcal{A}}$).

For any two sets $R, Q \subseteq \mathbb{Q}$ we write $R < Q$ if $r < q$ holds for all $r \in R$ and $q \in Q$. Given any real or rational number r, we denote the *integral part of* r by $\lfloor r \rfloor$, i.e. $\lfloor r \rfloor$ is the largest integer with $\lfloor r \rfloor \leq r$. Dually, $\lceil r \rceil$ addresses the smallest integer with $\lceil r \rceil \geq r$. By $\mathrm{fr}(r)$ we denote the *fractional part of* r, i.e. $\mathrm{fr}(r) := r - \lfloor r \rfloor$. Notice that $\mathrm{fr}(r)$ is always nonnegative, e.g. $\mathrm{fr}(3.71) = 0.71$, whereas $\mathrm{fr}(-3.71) = 0.29$. Given any tuple $\bar{r}$ of reals or rationals, we write $\mathrm{fr}(\bar{r})$ to address the corresponding tuple of fractional parts, i.e. $\mathrm{fr}\big(\langle r_1, \ldots, r_\mu \rangle\big) := \big\langle \mathrm{fr}(r_1), \ldots, \mathrm{fr}(r_\mu) \big\rangle$. We use the notation $\lfloor \bar{r} \rfloor$ and $\lceil \bar{r} \rceil$ in a component-wise fashion as well. Throughout Part II we shall use the usual notation for intervals of the number line: for example, with respect to the rational numbers $(-\infty, r]$ denotes the set $\{q \in \mathbb{Q} \mid q \leq r\}$ and $[r, s)$ denotes the set $\{q \in \mathbb{Q} \mid r \leq q < s\}$. It will always be clear from the current context what the underlying domain is. Finally, $\mathbb{Q}_{\geq 0}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{Z}_{\geq 0}$ address the sets of all nonnegative rational numbers, real numbers, and integers, respectively.

$R < Q$

$\lfloor r \rfloor$, $\mathrm{fr}(r)$

$\mathbb{Q}_{\geq 0}$, $\mathbb{R}_{\geq 0}$, $\mathbb{Z}_{\geq 0}$

Chapter 10

Decidable Fragments of Linear Rational Arithmetic with Uninterpreted Predicate Symbols

We have emphasized in Chapter 8 that the syntax of decidable first-order fragments combining arithmetic with uninterpreted predicate symbols has to be restricted considerably on both sides, the arithmetic part as well as the uninterpreted part. In the present chapter, we shall introduce and investigate two subfragments with a decidable satisfiability problem, both based on the Bernays–Schönfinkel–Ramsey fragment: *BSR with simple linear rational constraints — BSR(SLR)* — and *BSR with bounded difference constraints — BSR(BD).* We have already sketched the definitions in Chapter 8. The formal definitions are as follows.

Definition 10.0.1 (BSR with simple linear rational constraints — BSR(SLR))**.** *A* BSR(SLR) clause *has the form* $\Lambda \wedge \Gamma \to \Delta$*, where the conjunctions* Λ*,* Γ *and the disjunction* Δ *satisfy the following conditions:*

(i) Every atom in Λ *is an LRA atom of the form* $s \lhd t$ *or* $x \lhd t$ *or* $x \lhd y$ *where* s, t *are ground (i.e. variable-free) LRA terms,* $x, y \in \mathrm{Var}_{\mathbb{Q}}$*, and* $\lhd \in \{<, \leq, =, \neq, \geq, >\}$*.*

(ii) Every atom in Γ *and* Δ *is either an equation* $s \approx s'$ *over free-sort variables and constant symbols, or a non-equational atom* $P(s_1, \ldots, s_m)$ *that is well sorted and where the* s_i *range over base-sort variables, free-sort variables, and free-sort constant symbols.*

Definition 10.0.2 (BSR with bounded difference constraints — BSR(BD))**.** *A* BSR(BD) clause *has the form* $\Lambda \wedge \Gamma \to \Delta$*, where the conjunctions* Λ*,* Γ *and the disjunction* Δ *satisfy Condition (ii) of Definition 10.0.1, and every atom in* Λ *is an LRA atom of the form* $x \lhd c$*,* $x \lhd y$*, or* $x - y \lhd c$ *where* $c \in \mathbb{Z}$ *may be any integer (not a Skolem constant!),* $x, y \in \mathrm{Var}_{\mathbb{Q}}$ *are distinct, and* $\lhd \in \{<, \leq, =, \neq, \geq, >\}$*. Moreover, we require that whenever* Λ *contains an atom of the form* $x - y \lhd c$*, then* Λ *also contains LRA atoms* $c_x \leq x$*,* $x \leq d_x$*,* $c_y \leq y$*, and* $y \leq d_y$ *with* $c_x, d_x, c_y, d_y \in \mathbb{Z}$*. We shall refer to atoms of the form* $x - y \lhd c$ *as* difference constraints*.*

Limiting the right-hand sides of arithmetic atoms in BSR(BD) clauses to integers instead of rational numbers simplifies their treatment. This restriction does not severely restrict expressiveness, as long as we are only interested in the satisfiability problem. We could multiply all rational numbers in a BSR(BD) clause set with the least common multiple of their denominators and thus obtain an equisatisfiable clause set in which only integers occur.

In Chapter 9 we have argued that requiring the parts Γ and Δ of clauses to not contain any arithmetic terms apart from variables does not limit expressiveness. In order to simplify syntax even further, we often restrict our attention to clause sets in the following normal forms.

Definition 10.0.3 (BSR(SLR) normal form for clauses and clause sets)**.** *A BSR(SLR) clause* $\Lambda \wedge \Gamma \to \Delta$ *is in* BSR(SLR) normal form *if every variable that occurs in* Λ *also occurs in* Γ *or in* Δ*.*

A BSR(SLR) clause set N *is in* BSR(SLR) normal form *if the following conditions are met. All clauses in* N *are in normal form and pairwise variable disjoint. Moreover,* N *can be divided into two parts* $N_{\mathbb{Q}}$ *and* N_{BSR} *such that*

(a) every clause in $N_{\mathbb{Q}}$ *has the form* $\Lambda \to \mathtt{false}$*, i.e. the parts* Γ *and* Δ *are empty, and*

(b) for every clause $\Lambda \wedge \Gamma \to \Delta$ *in* N_{BSR} *either* Γ *or* Δ *is nonempty and any LRA atom* $s \lhd t$ *in* Λ *is such that* s *and* t *are either base-sort variables or Skolem constants, respectively.*

Moreover, we assume that N_{BSR} *contains at least one free-sort constant symbol.*

Definition 10.0.4 (BSR(BD) normal form for clauses and clause sets)**.** *A BSR(BD) clause* $\Lambda \wedge \Gamma \to \Delta$ *is in* BSR(BD) normal form *if every variable that occurs in* Λ *also occurs in* Γ *or in* Δ*. A BSR(BD) clause set* N *is in* BSR(BD) normal form *if all clauses in* N *are in normal form and pairwise variable disjoint. Moreover, we assume that* N *contains at least one free-sort constant symbol.*

We have already seen exemplary clause sets for both fragments in Example 8.0.3. The requirement that (implicitly universally quantified) variables in the Λ-part of a clause $\Lambda \wedge \Gamma \to \Delta$ have to occur in Γ or Δ or in both can be established by any procedure for eliminating existentially quantified variables in conjunctions of LRA atoms. Establishing the other requirements in Definitions 10.0.3 and 10.0.4 is straightforward. We shall postpone the proof of this claim for a little while (cf. Lemma 10.0.7), and first derive an auxiliary result that will be useful to analyze the blowup that we incur during the normal form transformation.

m, $\bar{\mathrm{x}}$
x_0, x_i
$\Lambda(x_0, \bar{\mathrm{x}})$

The following is standard methodology in the area of *difference logic*, see, e.g. Section 5.7 in [KS16], Section 2.1 in [CM06], or Section 11.4.5 in [BT18]. Let m be any positive integer. Let $\bar{\mathrm{x}}$ be any m-tuple of pairwise-distinct first-order variables $x_1, \ldots, x_m$ of sort $\mathbb{Q}$ and let x_0 be any first-order variable of sort $\mathbb{Q}$ that does not occur in $\bar{\mathrm{x}}$. Let $\Lambda(x_0, \bar{\mathrm{x}}) := x_0 = 0 \wedge \Lambda'(x_0, \bar{\mathrm{x}})$ be a conjunction where $\Lambda'(x_0, \bar{\mathrm{x}})$ is a conjunction of atoms of the form $x - y \leq c$ or $x - y < c$ with $x, y \in \bar{\mathrm{x}} \cup \{x_0\}$ and $c \in \mathbb{Z}$.

$\mathcal{G}(\Lambda)$

Definition 10.0.5 (Difference constraint graph $\mathcal{G}_\Lambda$, cf. Definition 5.17 and Excercise 5.16 in [KS16])**.** *The* difference constraint graph $\mathcal{G}(\Lambda)$ *is a weighted directed graph* $\langle V, E\rangle$ *with* $V = \{x_0, x_1, \ldots, x_m\}$ *and* $E \subseteq V \times V \times \mathbb{Q}$ *such that*

$\langle x, x, 0\rangle \in E$ *for all* $x \in V$*,*

$\langle x, y, c\rangle \in E$ *if and only if* Λ *contains the constraint* $x - y \leq c$*, and*

$\langle x, y, c - \delta\rangle \in E$ *if and only if* Λ *contains the constraint* $x - y < c$*,*

where we set $\delta := \frac{1}{2}(m+1)^{-1}$*.*

A *path* π in $\mathcal{G}_\Lambda$ is any finite, nonempty sequence $\langle x_{i_1}, x_{i_2}, c_1\rangle\langle x_{i_2}, x_{i_3}, c_2\rangle \ldots \langle x_{i_{\ell-1}}, x_{i_\ell}, c_\ell\rangle$ of edges from $\mathcal{G}_\Lambda$. We call π *simple*, if the indices $i_1, \ldots, i_{\ell-1}$ are pairwise distinct, i.e. π traverses every vertex in $\mathcal{G}_\Lambda$ at most once, except for the end point which may coincide with the starting point but does not have to. A *simple cycle* in $\mathcal{G}_\Lambda$ is any simple path whose start and end point coincide. The *length of a path* in $\mathcal{G}_\Lambda$ is the sum of the weights associated with the edges the path traverses. Notice that δ in Definition 10.0.5 is chosen such that the following property is satisfied. Let π be any simple path in $\mathcal{G}_\Lambda$. Let $c_1, \ldots, c_\ell$ be the weights associated with the edges π traverses. We have $\left(\sum_{1 \leq i \leq \ell} \lceil c_i \rceil\right) - 1 < \sum_{1 \leq i \leq \ell} c_i \leq \sum_{1 \leq i \leq \ell} \lceil c_i \rceil$.

Proposition 10.0.6 (cf. Theorem 1 in [CM06])**.** *Consider the difference constraint graph* $\mathcal{G}_\Lambda$ *and suppose that we have* $\mathbb{Q} \models \exists x_0 \bar{\mathrm{x}}.\, \Lambda(x_0, \bar{\mathrm{x}})$*. Then, for every pair* $x, y \in \bar{\mathrm{x}} \cup \{x_0\}$ *and every rational number* r *we have*

(a) $\mathbb{Q} \models \forall x_0 \bar{\mathrm{x}}.\, \Lambda(x_0, \bar{\mathrm{x}}) \to x - y \leq r$ *if and only if* y *is reachable from* x *in* $\mathcal{G}_\Lambda$ *and* $\lceil d_{x,y} \rceil \leq r$*, and*

(b) $\mathbb{Q} \models \forall x_0 \bar{\mathrm{x}}.\, \Lambda(x_0, \bar{\mathrm{x}}) \to x - y < r$ *if and only if* y *is reachable from* x *in* $\mathcal{G}_\Lambda$ *and we have either* $\lceil d_{x,y} \rceil < r$ *or* $d_{x,y} < \lceil d_{x,y} \rceil = r$*,*

where $d_{x,y}$ *is the length of a shortest simple path from* x *to* y *in* $\mathcal{G}_\Lambda$*.*

In fact, a variant of Proposition 10.0.6 yields a deterministic decision procedure for the sentence $\psi := \exists x_0 \bar{\mathrm{x}}.\, \Lambda(x_0, \bar{\mathrm{x}})$ under $\mathbb{Q}$ that runs in polynomial time [KS16, BT18]: ψ is satisfied by $\mathbb{Q}$ if and only if there is some simple cycle in $\mathcal{G}_\Lambda$ that has a negative length. In other words, we then have $\mathbb{Q} \models \forall x_0 \bar{\mathrm{x}}.\, \Lambda(x_0, \bar{\mathrm{x}}) \to x - x \leq -1$ for some $x \in \bar{\mathrm{x}} \cup \{x_0\}$.

Next, we prove the existence of BSR(SLR) and BSR(BD) normal forms.

Lemma 10.0.7. *For every BSR(SLR) (or BSR(BD)) clause set* N *there is an equisatisfiable BSR(SLR) (or BSR(BD)) clause set* N' *in BSR(SLR) normal form (BSR(BD) normal form) such that*

(a) the length of N' *is at most exponential in the length of* N*,*

(b) for any clause C *in* N' *the number of variables occurring in* C *is not larger than the number of variables occurring in any clause in* N*,*

(c) if N *is a BSR(SLR) clause set, the number of distinct rational numbers and Skolem constants occurring in* N' *is linear in the length of* N*,*

(d) if N *is a BSR(BD) clause set, then*

  (d.1) the number of clauses in N' *grows at most exponentially in the number of atoms* $s \neq t$ *occurring in any clause in* N*,*

  (d.2) the length of any clause in N' *is at most polynomial in the length of the longest clause in* N*,*

  (d.3) every free-sort Skolem constant occurring in N' *also occurs in* N*, and*

  (d.4) the absolute value of any integer in N' *is linear in* $\kappa \cdot \lambda$*, where* κ *is the smallest positive integer that is larger than the absolute value of any integer occurring in* N*, and* λ *is the smallest positive integer that is larger than the maximal number of universally quantified variables occurring in any clause in* N*.*

Proof sketch. We start with the BSR(SLR) case. First, we show how make sure that every base-sort variable that occurs in Λ in a clause $\Lambda \wedge \Gamma \to \Delta$ also occurs in Γ or in Δ. Consider any BSR(SLR) clause $\Lambda \wedge \Gamma \to \Delta$ and let $\bar{\mathrm{x}}$ be some nonempty tuple of base-sort variables that occur in Λ but neither in Γ nor in Δ. Recall that all variables in clauses are implicitly universally quantified. We observe that $\forall \bar{\mathrm{x}}.\, (\Lambda \wedge \Gamma \to \Delta)$ is equivalent to $(\exists \bar{\mathrm{x}}.\, \Lambda) \wedge \Gamma \to \Delta$. Since Λ is a conjunction of LRA atoms, we may apply virtual substitution (cf. Section 7.1, page 185) to eliminate the quantifier block $\exists \bar{\mathrm{x}}$ and compute some disjunction of conjunctions of LRA atoms $\bigvee_i \Lambda'_i$ that is $\mathbb{Q}$-equivalent to $\exists \bar{\mathrm{x}}.\, \Lambda$. Then, the clause $(\exists \bar{\mathrm{x}}.\, \Lambda) \wedge \Gamma \to \Delta$ is equivalent to the conjunction of clauses $\bigwedge_i (\Lambda'_i \wedge \Gamma \to \Delta)$. The length of $\bigwedge_i (\Lambda'_i \wedge \Gamma \to \Delta)$ is at most exponential in the length of $\Lambda \wedge \Gamma \to \Delta$ (cf. Theorem 3.7 in [LW93]), the length of each Λ_i is at most linear in the length of Λ, and the set of variables occurring in any Λ'_i is a subset of the variables occurring freely in $\exists \bar{\mathrm{x}}.\, \Lambda$. In BSR(SLR) clauses the used elimination sets contain only testpoints of the form t, $t + \varepsilon$, or $-\infty$, where t is some LRA term occurring in N in some atom $x \lhd t$. Virtually substituting such a testpoint in any arithmetic atom that is admitted in BSR(SLR) yields again an atom admitted in BSR(SLR).

$\Lambda \wedge \Gamma \to \Delta$, $\bar{\mathrm{x}}$ Λ'_i

Next, we describe how to modify N in such a way that it can be partitioned into $N_\mathbb{Q}$ and N_{BSR} as required in Definition 10.0.3. Clauses of the form $s \lhd t \wedge \Lambda' \wedge \Gamma \to \Delta$, where t is neither a variable nor a Skolem constant, are replaced — under preservation of (un)satisfiability — with two clauses $t \neq c \to \texttt{false}$ (which is equivalent to $t = c$) and $s \lhd c \wedge \Lambda' \wedge \Gamma \to \Delta$ for some fresh uninterpreted

constant symbol c of sort $\mathbb{Q}$. Doing this exhaustively for all clauses with nonempty part Γ or Δ leads to the desired partition of N in $N_{\mathbb{Q}}$ and N_{BSR}.

Now we treat the BSR(BD) case. Again, we first show how to make sure that every base-sort variable that occurs in Λ in a clause $\Lambda \wedge \Gamma \to \Delta$ also occurs in Γ or in Δ. Clauses of the form $s \neq t \wedge \Lambda' \wedge \Gamma \to \Delta$ are equivalently replaced with two clauses $s < t \wedge \Lambda' \wedge \Gamma \to \Delta$ and $s > t \wedge \Lambda' \wedge \Gamma \to \Delta$. We do this exhaustively for all atoms $s \neq t$ that contain at least one variable not occurring in the Γ or Δ part of the respective clause. In the worst case, treating a clause in N in this way produces 2^k clauses if the original clause contains k atoms $s \neq t$ that need to be replaced.

C, Λ', Λ, Γ, Δ, $\bar{\mathrm{x}}$, $\bar{\mathrm{v}}$

Consider any BSR(BD) clause $C := \Lambda' \wedge \Lambda \wedge \Gamma \to \Delta$ where every atom in Λ' contains a variable x that does not occur in Λ, Γ, and Δ. Let $\bar{\mathrm{x}}$ be some tuple listing all these variables exactly once and let $\bar{\mathrm{v}}$ be some tuple listing all the other variables occurring in C. We assume that Λ' does not contain any atoms of the from $s \neq t$. Moreover, we assume that all atoms $s = t$ in Λ' have been replaced with conjunctions $s \leq t \wedge t \leq s$. We observe that $\forall \bar{\mathrm{x}}.\, (\Lambda'(\bar{\mathrm{x}}, \bar{\mathrm{v}}) \wedge \Lambda(\bar{\mathrm{v}}) \wedge \Gamma(\bar{\mathrm{v}}) \to \Delta(\bar{\mathrm{v}}))$ is equivalent to $\big(\exists \bar{\mathrm{x}}.\, \Lambda'(\bar{\mathrm{x}}, \bar{\mathrm{v}})\big) \wedge \Lambda(\bar{\mathrm{v}}) \wedge \Gamma(\bar{\mathrm{v}}) \to \Delta(\bar{\mathrm{v}})$. Since $\Lambda'(\bar{\mathrm{x}}, \bar{\mathrm{v}})$ is a conjunction of LRA atoms, we may apply the *Fourier-Motzkin elimination procedure* to eliminate the variables $\bar{\mathrm{x}}$ in $\big(\exists \bar{\mathrm{x}}.\, \Lambda'(\bar{\mathrm{x}}, \bar{\mathrm{v}})\big)$ one by one.

x, Λ'_0, Λ'_1, Λ'_2

Consider any $x \in \bar{\mathrm{x}}$. In order to eliminate x from $\exists x.\, \Lambda'(\bar{\mathrm{x}}, \bar{\mathrm{v}})$, we proceed as follows. Let $\Lambda'_0, \Lambda'_1, \Lambda'_2$ be the shortest conjunctions satisfying the following properties:

(i) every atom from Λ' that does not contain x occurs in Λ'_0,

(ii) for every atom in Λ' that contains x there is a $\mathbb{Q}$-equivalent atom in $\Lambda'_1 \wedge \Lambda'_2$,

(iii) every atom in Λ'_1 has the form $s \leq x$ or $s < x$ where s is either an integer, a variable, or an LRA term $y + c$ for some variable $y \in \bar{\mathrm{x}} \cup \bar{\mathrm{v}}$ of sort $\mathbb{Q}$ and some integer c, and

(iv) every atom in Λ'_2 has the form $x \leq t$ or $x < t$ where t is either an integer, a variable, or an LRA term $y + c$ for some variable $y \in \bar{\mathrm{x}} \cup \bar{\mathrm{v}}$ of sort $\mathbb{Q}$ and some integer c.

Λ''

Let Λ'' be the conjunction of the following set of atoms

$$\left\{ s < t \;\middle|\; (s \lhd_1 x) \in \Lambda'_1 \text{ and } (x \lhd_2 t) \in \Lambda'_2 \text{ where at least one of } \lhd_1, \lhd_2 \text{ is the strict } < \right\}$$
$$\cup \left\{ s \leq t \;\middle|\; (s \leq x) \in \Lambda'_1 \text{ and } (x \leq t) \in \Lambda'_2 \right\}.$$

$\bar{\mathrm{x}}'$

Let $\bar{\mathrm{x}}' := \bar{\mathrm{x}} \setminus \{x\}$. It is well known that the two formulas $\exists x.\, \Lambda'_1(x, \bar{\mathrm{x}}', \bar{\mathrm{v}}) \wedge \Lambda'_2(x, \bar{\mathrm{x}}', \bar{\mathrm{v}})$ and $\Lambda''(\bar{\mathrm{x}}', \bar{\mathrm{v}})$ are $\mathbb{Q}$-equivalent (see, e.g. [Sch99], Section 12.2). Hence, $\exists x.\, \Lambda'(x, \bar{\mathrm{x}}', \bar{\mathrm{v}})$ can be replaced with the $\mathbb{Q}$-equivalent formula $\Lambda'_0(\bar{\mathrm{x}}', \bar{\mathrm{v}}) \wedge \Lambda''(\bar{\mathrm{x}}', \bar{\mathrm{v}})$. Concerning the atoms in $\Lambda''(\bar{\mathrm{x}}, \bar{\mathrm{v}})$ we find that every atom therein can be transformed into an equivalent atom of the form $y \lhd c$, $y \lhd z$, or $y - z \lhd c$ where $y, z \in \bar{\mathrm{x}}' \cup \bar{\mathrm{v}}$, c is some integer, and $\lhd \in \{<, \leq, \geq, >\}$. As we need to keep at most $4 \cdot |\bar{\mathrm{x}}' \cup \bar{\mathrm{v}}| + 8 \cdot |\bar{\mathrm{x}}' \cup \bar{\mathrm{v}}|^2$ of these atoms — at most one atom $y \lhd c$ for each pair $y, \lhd$ and at most two atoms $y - z \lhd d$ and $z - y \lhd e$ for every triple $y, z, \lhd$ —, we may assume that the length of $\Lambda''(\bar{\mathrm{x}}', \bar{\mathrm{v}})$ is at most polynomial in the number of variables in $\bar{\mathrm{x}}', \bar{\mathrm{v}}$.

$\Lambda'''(\bar{\mathrm{v}})$

We apply the described elimination procedure to eliminate the other variables in $\bar{\mathrm{x}}$ as well, in a variable-by-variable fashion. Hence, the final conjunction $\Lambda'''(\bar{\mathrm{v}})$ contains at most $4 \cdot |\bar{\mathrm{v}}| + 8 \cdot |\bar{\mathrm{v}}|^2$ atoms, and we replace the clause $C(\bar{\mathrm{x}}, \bar{\mathrm{v}})$ in N with the equivalent clause $\Lambda'''(\bar{\mathrm{v}}) \wedge \Lambda(\bar{\mathrm{v}}) \wedge \Gamma(\bar{\mathrm{v}}) \to \Delta(\bar{\mathrm{v}})$.

$\Lambda_{\mathrm{diff}}(\bar{\mathrm{x}}, \bar{\mathrm{v}})$

In addition, we can bound the absolute value of the integers occurring in Λ''' as follows. It is easy to verify that we can transform $\Lambda(\bar{\mathrm{x}}, \bar{\mathrm{v}})$ into a $\mathbb{Q}$-equivalent conjunction $\Lambda_{\mathrm{diff}}(\bar{\mathrm{x}}, \bar{\mathrm{v}})$ of difference constraints in the sense of Definition 10.0.5 and Proposition 10.0.6 (see the paragraph preceding Definition 10.0.5). We have mentioned right after Proposition 10.0.6 that we can check in polynomial time whether $\exists \bar{\mathrm{x}} \bar{\mathrm{v}}.\, \Lambda_{\mathrm{diff}}(\bar{\mathrm{x}}, \bar{\mathrm{v}})$ is satisfied under $\mathbb{Q}$. In the opposite case, Λ''' can in fact be replaced by `false`. Henceforth, we assume that $\mathbb{Q} \models \exists \bar{\mathrm{x}} \bar{\mathrm{v}}.\, \Lambda_{\mathrm{diff}}(\bar{\mathrm{x}}, \bar{\mathrm{v}})$. Since $\Lambda'''(\bar{\mathrm{v}})$ is the result of applying Fourier-Motzkin elimination to $\exists \bar{\mathrm{x}}.\, \Lambda'(\bar{\mathrm{x}}, \bar{\mathrm{v}})$, we observe that for every atom of the form $u - v \leq c$ occurring in $\Lambda'''(\bar{\mathrm{v}})$ we have $\mathbb{Q} \models \forall \bar{\mathrm{x}} \bar{\mathrm{v}}.\, \Lambda_{\mathrm{diff}}(\bar{\mathrm{x}}, \bar{\mathrm{v}}) \to u - v \leq c$. Let κ be the smallest positive integer

κ

that is larger than the absolute value of any integer occurring in Λ'. Then, by Proposition 10.0.6, we observe $c \geq -\kappa \cdot (|\bar{\mathrm{x}} \cup \bar{\mathrm{v}}| + 1)$ and, in addition, that there exists some integer k satisfying the following properties:

(1) $-\kappa \cdot (|\bar{\mathrm{x}} \cup \bar{\mathrm{v}}| + 1) \ \leq \ k \ \leq \ \kappa \cdot (|\bar{\mathrm{x}} \cup \bar{\mathrm{v}}| + 1)$, and

(2) $\mathbb{Q} \models \forall \bar{\mathrm{x}} \bar{\mathrm{v}}.\, \Lambda'(\bar{\mathrm{x}}, \bar{\mathrm{v}}) \rightarrow u - v \triangleleft k$.

This means, if $c > \kappa \cdot (|\bar{\mathrm{x}} \cup \bar{\mathrm{v}}| + 1)$, then we can replace $u - v \leq c$ in Λ''' with the atom $u - v \leq k$, which subsumes the former. Using similar arguments we can show the same for other atoms occurring in Λ'''. Consequently, we may assume that Λ''' contains only integers whose absolute value is linear in $\kappa \cdot (|\bar{\mathrm{x}} \cup \bar{\mathrm{v}}| + 1)$. □

The main result of the present Chapter is that satisfiability of finite BSR(SLR) clause sets and finite BSR(BD) clause sets is decidable, respectively (Theorems 10.2.14 and 10.4.10). The proof technique is very similar for the two fragments. It is partially based on methods from Ramsey theory, which will be briefly introduced in the following section.

10.1 Basic Tools from Ramsey Theory

In the present section we establish two technical results based on methods usually applied in Ramsey theory. We shall use these results later on to prove the existence of models of a particular kind for finite and satisfiable BSR(SLR) or BSR(BD) clause sets. These models meet certain uniformity conditions. In order to construct them, we rely on the existence of certain finite subsets of $\mathbb{Q}$ that are used to construct prototypical tuples of rational numbers. These finite subsets, in turn, have to behave nicely as well, since rational tuples that are not distinguishable by BSR(SLR) or BSR(BD) clauses are required to have certain uniformity properties.

A tuple $\langle r_1, \ldots, r_m \rangle \in \mathbb{Q}^m$ is called *ascending* if $r_1 < \ldots < r_m$. A *coloring* is a mapping $\chi : S \rightarrow \mathcal{C}$ for any set S and any finite set $\mathcal{C}$. For the most basic result of this section (Lemma 10.1.1), we consider an arbitrary coloring χ of m-tuples of rational numbers and stipulate the existence of a finite subset $Q \subseteq \mathbb{Q}$ of a given cardinality n such that all ascending m-tuples of elements from Q are assigned the same color by χ. We call such a set Q *uniformly colored*.

ascending tuples, coloring

uniformly colored sets

Lemma 10.1.1. *Let $n, m > 0$ be positive integers. Let $\chi : \mathbb{Q}^m \rightarrow \mathcal{C}$ be any coloring. There is some positive integer $\widehat{n}$ such that for every set $R \subseteq \mathbb{Q}$ with $|R| \geq \widehat{n}$ — i.e. R needs to be sufficiently large — there exists a subset $Q \subseteq R$ of cardinality n such that all ascending tuples $\langle r_1, \ldots, r_m \rangle \in Q^m$ are assigned the same color by χ.*

Proof. This proof is an adaptation of the proof of Ramsey's Theorem on page 7 in [GRS90]. For $n < m$ the lemma is trivially satisfied, since in this case Q^m cannot contain any ascending tuple. Hence, we assume $n \geq m$. In order to avoid technical difficulties when defining the sequence of elements $s_{m-1}, s_m, s_{m+1}, \ldots$ below, we assume for the rest of the proof that R is finite but sufficiently large. This assumption does not pose a restriction, as we could always consider a sufficiently large finite subset of R, if R were to be infinite.

We proceed by induction on $m \geq 1$. The base case $m = 1$ is easy, since χ can assign only finitely many colors to elements in R and thus some color must be assigned at least $\lfloor \frac{|R|}{|\mathcal{C}|} \rfloor$ times. Hence, if R contains at least $n|\mathcal{C}|$ elements, we find a uniformly colored subset Q of size n. Suppose $m > 1$. At first, we pick the $m-2$ smallest rational numbers $s_1 < \ldots < s_{m-2}$ from R and set $S_{m-2} := R \setminus \{s_1, \ldots, s_{m-2}\}$. Thereafter, we simultaneously construct two *sufficiently long but finite* sequences $s_{m-1}, s_m, s_{m+1}, \ldots$ and $S_{m-1}, S_m, S_{m+1}, \ldots$ as follows:

$s_1, \ldots, s_{m-2}$, S_{m-2}

Given S_i, we define s_{i+1} to be the smallest rational number in S_i.

s_{i+1}

Given S_i and the element s_{i+1}, we define an equivalence relation $\sim_i$ on the set $S'_i := S_i \setminus \{s_{i+1}\}$ so that $s \sim_i s'$ holds if and only if for every sequence of indices $j_1, \ldots, j_{m-1}$ with $1 \leq j_1 < \ldots < j_{m-1} \leq i+1$, we have $\chi(s_{j_1}, \ldots, s_{j_{m-1}}, s) = \chi(s_{j_1}, \ldots, s_{j_{m-1}}, s')$. This equivalence relation

$\sim_i$, S'_i

partitions S_i' into at most $|\mathcal{C}|^{\binom{i+1}{m-1}}$ equivalence classes. We choose one such class with largest cardinality to be S_{i+1}.

S_{i+1}

By construction of the sequence $s_1, s_2, s_3, \ldots$, we must have $\chi(s_{j_1}, \ldots, s_{j_{m-1}}, s_K) = \chi(s_{j_1}, \ldots, s_{j_{m-1}}, s_{K'})$ for every sequence of indices $j_1 < \ldots < j_{m-1}$ and all indices $K, K' \geq j_{m-1} + 1$. Notice that this covers all ascending m-tuples in $\{s_1, s_2, s_3, \ldots\}^m$ starting with $s_{j_1}, \ldots, s_{j_m-1}$, i.e. they all share the same color. We now define a new coloring $\chi' : \{s_1, s_2, s_3, \ldots\}^{m-1} \to \mathcal{C}$ so that $\chi'(s_{j_1}, \ldots, s_{j_{m-1}}) := \chi(s_{j_1}, \ldots, s_{j_{m-1}}, s_{j_{m-1}+1})$ for every sequence of indices $j_1 < \ldots < j_{m-1}$ (in case of j_{m-1} being the index of the last element in the sequence $s_1, s_2, s_3, \ldots$, $\chi'(s_{j_1}, \ldots, s_{j_{m-1}})$ shall be an arbitrary color from $\mathcal{C}$). By induction, there exists a subset $Q \subseteq \{s_1, s_2, s_3, \ldots\}$ of cardinality n, such that every ascending $(m-1)$-tuple $\bar{r} \in Q^{m-1}$ is colored the same by χ'. The definition of χ' entails that now all ascending m-tuples $\bar{r}' \in Q^m$ are colored the same by χ. Hence, Q is the sought set. □

χ'

Q

Based on Lemma 10.1.1, one can derive similar results for more structured ways of coloring tuples of rational numbers. We shall employ such a structured coloring when proving that the satisfiability problem for finite BSR(SLR) clause sets is decidable. More precisely, the proof of Lemma 10.2.9 will rely on such a result, namely Lemma 10.1.4. But before we formulate and prove this lemma, we present two auxiliary results.

Lemma 10.1.2. *Let $n, m, p > 0$ be positive integers and let $\chi : \mathbb{Q}^{mp} \to \mathcal{C}$ be an arbitrary coloring. Let $R_1, \ldots, R_p$ be* sufficiently large *but finite subsets of $\mathbb{Q}$. There exist subsets $Q_1 \subseteq R_1, \ldots, Q_p \subseteq R_p$, each of cardinality n and there is some color $C \in \mathcal{C}$, such that for all ascending m-tuples $\bar{r}_1 \in Q_1^m, \ldots, \bar{r}_p \in Q_p^m$ we have $\chi(\bar{r}_1, \ldots, \bar{r}_p) = C$.*

Proof. This proof is an adaptation of the proof of Theorem 5 on page 113 in [GRS90]. As in the proof of Lemma 10.1.1, we assume $n \geq m$. We proceed by induction on $p \geq 1$. The case $p = 1$ is covered by Lemma 10.1.1. Suppose $p > 1$. We define an equivalence relation $\sim_p$ over the set R_p^m so that $\bar{s} \sim_p \bar{s}'$ holds if and only if for all ascending tuples $\bar{r}_1 \in R_1^m, \ldots, \bar{r}_{p-1} \in R_{p-1}^m$ the colors $\chi(\bar{r}_1, \ldots, \bar{r}_{p-1}, \bar{s})$ and $\chi(\bar{r}_1, \ldots, \bar{r}_{p-1}, \bar{s}')$ are identical. This equivalence relation induces at most $|\mathcal{C}|^{\binom{|R_1|}{m} \cdot \ldots \cdot \binom{|R_{p-1}|}{m}}$ equivalence classes over R_p^m. It thus induces a coloring of $\chi' : R_p^m \to \mathcal{C}_p'$ where $\mathcal{C}_p'$ contains one color for each of these equivalence classes. By virtue of Lemma 10.1.1, we can construct a subset $Q_p \subseteq R_p$ with n elements such that all ascending m-tuples $\bar{r} \in Q_p^m$ are colored identically by χ'. Let the coloring χ'' be defined by $\chi''(\bar{r}_1, \ldots, \bar{r}_{p-1}) := \chi(\bar{r}_1, \ldots, \bar{r}_{p-1}, \bar{s})$ for some fixed ascending m-tuple $\bar{s} \in Q_p^m$. By induction, we find subsets $Q_1 \subseteq R_1, \ldots, Q_{p-1} \subseteq R_{p-1}$, each containing n elements, such that for all ascending m-tuples $\bar{r}_1 \in R_1^m, \ldots, \bar{r}_{p-1} \in R_{p-1}^m$ the colors $\chi''(\bar{r}_1, \ldots, \bar{r}_{p-1})$ are identical. But then, the definition of χ'' and χ' entail that the sets $Q_1, \ldots, Q_p$ satisfy the requirements posed by the lemma. □

Recall that we write $[K]$ to address the set $\{1, \ldots, K\}$ for any positive integer $K > 0$.

Lemma 10.1.3. *Let $n, m, p > 0$ be positive integers, let $K \geq 0$ be a nonnegative integer and let $\chi : \mathbb{Q}^m \to \mathcal{C}$ be an arbitrary coloring. Let $R_1, \ldots, R_p$ be* sufficiently large *but finite subsets of $\mathbb{Q}$. Let $q_1, \ldots, q_K$ be fixed rational numbers. Let $\rho : [m] \to [p + K] \times [m]$ be some mapping such that $\rho(i) = \langle K, \ell \rangle$ with $K > p$ implies $\ell = 1$.*

There exist subsets $Q_1 \subseteq R_1, \ldots, Q_p \subseteq R_p$, each of cardinality n, and there exists some color $C \in \mathcal{C}$ such that for all ascending tuples

$$\begin{aligned}
\bar{r}_1 &= \langle r_{\langle 1,1 \rangle}, \ldots, r_{\langle 1,m \rangle} \rangle \in Q_1^m \\
&\vdots \\
\bar{r}_p &= \langle r_{\langle p,1 \rangle}, \ldots, r_{\langle p,m \rangle} \rangle \in Q_p^m \\
\bar{r}_{p+1} &= \langle r_{\langle p+1,1 \rangle} \rangle := \langle q_1 \rangle \\
&\vdots \\
\bar{r}_{p+K} &= \langle r_{\langle p+K,1 \rangle} \rangle := \langle q_K \rangle
\end{aligned}$$

we have $\chi(\bar{r}_{\rho(1)}, \ldots, \bar{r}_{\rho(m)}) = C$.

Proof. We again assume $n \geq m$. We define a new coloring $\chi' : \mathbb{Q}^{mp} \to \mathcal{C}$ by

$$\chi'(r_{\langle 1,1\rangle}, \ldots, r_{\langle 1,m\rangle}, \ldots, r_{\langle p,1\rangle}, \ldots, r_{\langle p,m\rangle}) := \chi(r_{\rho(1)}, \ldots, r_{\rho(m)})$$

for every mp-tuple $\langle \bar{r}_1, \ldots, \bar{r}_p\rangle \in R_1^m \times \ldots \times R_p^m$ with ascending $\bar{r}_1, \ldots, \bar{r}_p$. By Lemma 10.1.2, there exist subsets $Q_1 \subseteq R_1, \ldots, Q_p \subseteq R_p$, each with n elements, such that for all ascending tuples $\bar{r}_1 \in Q_1^m, \ldots, \bar{r}_p \in Q_p^m$ the colors $\chi'(\bar{r}_1, \ldots, \bar{r}_p)$ are the same. By definition of χ', the sets $Q_1, \ldots, Q_p$ satisfy the requirements of the lemma. □

Now we have the right tools at hand to prove Lemma 10.1.4

Lemma 10.1.4. *Let $n, m, p > 0$ be positive integers, let $K \geq 0$ be a nonnegative integer and let $\chi : \mathbb{Q}^m \to \mathcal{C}$ be an arbitrary coloring. Let $R_1, \ldots, R_p$ be* sufficiently large *but finite subsets of $\mathbb{Q}$. Let $q_1, \ldots, q_K$ be fixed rational numbers. Let $\rho_1, \ldots, \rho_L$ be some enumeration of all mappings $\rho_j : [m] \to [p + K] \times [m]$ for which $\rho_j(i) = \langle K, \ell\rangle$ with $K > p$ entails $\ell = 1$. Then, there exist subsets $Q_1 \subseteq R_1, \ldots, Q_p \subseteq R_p$, each of cardinality n, such that for all ascending tuples*

$$\begin{aligned}
\bar{r}_1, \bar{r}'_1 &\in Q_1^m \\
&\vdots \\
\bar{r}_p, \bar{r}'_p &\in Q_p^m \\
\bar{r}_{p+1} &:= \langle r_{p+1,1}\rangle := \langle q_1\rangle \\
&\vdots \\
\bar{r}_{p+K} &:= \langle r_{p+K,1}\rangle := \langle q_K\rangle
\end{aligned}$$

and every index j, $1 \leq j \leq L$, we have

$$\chi\big(r_{\rho_j(1)}, \ldots, r_{\rho_j(m)}\big) = \chi\big(r'_{\rho_j(1)}, \ldots, r'_{\rho_j(m)}\big).$$

Proof. We again assume $n \geq m$. We construct sequences of subsets $S_{\ell,0} \supseteq \ldots \supseteq S_{\ell,L}$ for every ℓ, $1 \leq \ell \leq p$, such that

$S_{\ell,0} = R_\ell$, and

$S_{\ell,j+1} \subseteq S_{\ell,j}$ is a subset of *sufficient cardinality* that is constructed by application of Lemma 10.1.3 for $\rho := \rho_{j+1}$, i.e. for all ascending tuples

$$\begin{aligned}
\langle s_{\langle 1,1\rangle}, \ldots, s_{\langle 1,m\rangle}\rangle &\in S_{1,j+1}^m \\
&\vdots \\
\langle s_{\langle p,1\rangle}, \ldots, s_{\langle p,m\rangle}\rangle &\in S_{p,j+1}^m
\end{aligned}$$

the colors $\chi(\bar{s}_{\rho_{j+1}(1)}, \ldots, \bar{s}_{\rho_{j+1}(m)})$ are the same.

Then the sets $S_{1,L}, \ldots, S_{p,L}$ are the sought $Q_1, \ldots, Q_p$. □

10.2 The Bernays–Schönfinkel–Ramsey Fragment with Simple Linear Rational Constraints is Decidable

For the rest of the present section we fix two positive integers $m, m' > 0$ and some finite BSR(SLR) clause set N in BSR(SLR) normal form. For the sake of simplicity, we assume that all uninterpreted predicate symbols P occurring in N have the sort $P : \mathcal{S}^{m'} \times \mathbb{Q}^m$. This assumption does not limit expressiveness, as the arity of a predicate symbol P can easily be increased in an (un)satisfiability-preserving way by padding the occurring atoms with additional arguments. For instance, every

N, m, m'

occurrence of an atom $P(t_1, \ldots, t_m)$ can be replaced with $P(t_1, \ldots, t_m, v, \ldots, v)$ for some fresh first-order variable v that is added sufficiently often as argument.

Given the BSR(SLR) clause set N, every structure $\mathcal{A}$ induces a partition of $\mathbb{Q}$ into finitely many intervals: the rational numbers occurring in N together with the interpretations of all the Skolem constants c occurring in N yield point intervals that are interspersed with and enclosed by open intervals.

Definition 10.2.1 ($\mathcal{A}$-induced partition of $\mathbb{Q}$)**.** *Let $\mathcal{A}$ be any structure and let $r_1, \ldots, r_k$ be an enumeration of all the values in the set $\{c^{\mathcal{A}} \mid c \in \text{consts}(N)$ is of sort $\mathbb{Q}\}$ in ascending order. By $\mathcal{J}_{\mathcal{A}}$ we denote the following partition of $\mathbb{Q}$:*

$\mathcal{J}_{\mathcal{A}}$

$$\mathcal{J}_{\mathcal{A}} := \big\{(-\infty, r_1), \{r_1\}, (r_1, r_2), \{r_2\}, \ldots, (r_{k-1}, r_k), \{r_k\}, (r_k, +\infty)\big\},$$

where the sets $\{r_i\}$ represent point intervals, i.e. closed intervals containing exactly one value r_i.

The idea underlying the following equivalence relation is that equivalent tuples are indistinguishable with respect to the arithmetic atoms that we allow in the BSR(SLR) clause set N.

Definition 10.2.2 ($\mathcal{J}_{\mathcal{A}}$-equivalence, $\sim_{\mathcal{J}_{\mathcal{A}}}$)**.** *Let $\mathcal{A}$ be any structure and let k be any positive integer. We call two k-tuples $\bar{r}, \bar{q} \in \mathbb{Q}^k$ $\mathcal{J}_{\mathcal{A}}$-*equivalent *if and only if*

(i) for every i, $1 \leq i \leq k$, and every interval $J \in \mathcal{J}_{\mathcal{A}}$ we have $r_i \in J$ if and only if $q_i \in J$ and

(ii) for all i, j, $1 \leq i, j \leq k$ we have $r_i < r_j$ if and only if $q_i < q_j$.

The induced equivalence relation over tuples of positive length is denoted by $\sim_{\mathcal{J}_{\mathcal{A}}}$. $\sim_{\mathcal{J}_{\mathcal{A}}}$

For every positive k the relation $\sim_{\mathcal{J}_{\mathcal{A}}}$ induces only finitely many equivalence classes over the set of all k-tuples over the rationals.

Example 10.2.3. *Consider an exhaustively Skolemized variant of the sentence φ_1 from Example 8.0.3:*

$$\begin{aligned}\varphi_{\text{Sk}} := \forall xy.\, \big(3 < c_1 < c_2 < \tfrac{31}{5}\big) \wedge \big(3 \leq x < c_1 \;\wedge\; 3 < y < c_2 \;\rightarrow\; P(x,y)\big)\\ \wedge \big(c_1 \leq x < y \;\wedge\; y < \tfrac{31}{5} \;\rightarrow\; P(x,y)\big)\,.\end{aligned}$$

Although we use convenient notation, the sentence essentially meets the syntax of BSR(SLR). Let $\mathcal{A}$ be any structure satisfying $3 < c_1^{\mathcal{A}} < c_2^{\mathcal{A}} < \frac{31}{5}$. Then, the partition $\mathcal{J}_{\mathcal{A}}$ of $\mathbb{Q}$ is given by

$$\mathcal{J}_{\mathcal{A}} = \big\{(-\infty, 3), \{3\}, (3, c_1^{\mathcal{A}}), \{c_1^{\mathcal{A}}\}, (c_1^{\mathcal{A}}, c_2^{\mathcal{A}}), \{c_2^{\mathcal{A}}\}, (c_2^{\mathcal{A}}, \tfrac{31}{5}), \{\tfrac{31}{5}\}, (\tfrac{31}{5}, +\infty)\big\}\,.$$

Figure 10.1 illustrates the equivalence relation $\sim_{\mathcal{J}_{\mathcal{A}}}$ induced by $\mathcal{J}_{\mathcal{A}}$ over the two-dimensional rational plane.
Obviously, the number of equivalence classes is finite: there are 91 classes in the quotient set $\mathbb{Q}^2/_{\sim_{\mathcal{J}_{\mathcal{A}}}}$.

Proposition 10.2.4. *Any equivalence relation $\sim_{\mathcal{J}_{\mathcal{A}}}$ in accordance with Definition 10.2.2 induces finitely many equivalence classes over the set $\mathbb{Q}^k$.*

Proof. Given any k-tuple $\bar{r}$, every component belongs to exactly one of the intervals in $\mathcal{J}_{\mathcal{A}}$, and if multiple such components stem from the same interval, there are only finitely many possibilities for their ordering relative to one another. □

We intend to show that, if N is satisfiable, then there is some model $\mathcal{A}$ for N which does not distinguish between different $\mathcal{J}_{\mathcal{A}}$-equivalent tuples. First, we need some notion that reflects how the structure $\mathcal{A}$ treats a given tuple $\bar{r} \in \mathbb{Q}^m$. This role will be taken by the coloring $\chi_{\mathcal{A}}$, which maps $\bar{r}$ to a set of expressions of the form $P\bar{\mathsf{a}}$, where P is some predicate symbol occurring in N and $\bar{\mathsf{a}}$ is an m'-tuple of domain elements from $\mathcal{S}^{\mathcal{A}}$. The presence of $P\bar{\mathsf{a}}$ in the set $\chi_{\mathcal{A}}(\bar{r})$ indicates that $\mathcal{A}$ interprets P in such a way that $P^{\mathcal{A}}$ contains the tuple $\langle \bar{\mathsf{a}}, \bar{r} \rangle$. In this sense, $\chi_{\mathcal{A}}(\bar{r})$ comprises all the relevant information that $\mathcal{A}$ contains regarding the tuple $\bar{r}$.

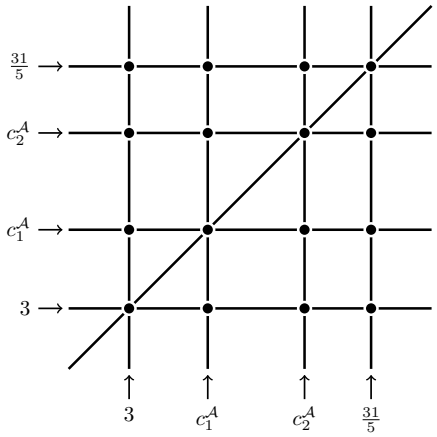


Figure 10.1: Partition of the two-dimensional rational plane into equivalence classes induced by $\sim_{\mathcal{J}_{\mathcal{A}}}$. Every dot, line segment, rectangular white area, and triangular white area represents an equivalence class. The open white areas represent open sets that stretch out towards infinity. The same holds true for the fringe line segments.

Definition 10.2.5 ($\mathcal{A}$-coloring $\chi_{\mathcal{A}}$)**.** *Let $c_1, \ldots, c_n$ be an enumeration of all constant symbols of sort $\mathcal{S}$ that occur in N. Given any structure $\mathcal{A}$, let $\widehat{\mathcal{S}} := \{\mathsf{a} \in \mathcal{S}^{\mathcal{A}} \mid \mathsf{a} = c_i^{\mathcal{A}} \text{ for some } c_i\}$. An $\mathcal{A}$-color is any set of expressions of the form $P\bar{\mathsf{a}}$ where P is some uninterpreted predicate symbol occurring in N and $\bar{\mathsf{a}}$ is some m'-tuple over the set $\widehat{\mathcal{S}}$. The $\mathcal{A}$-coloring of $\mathbb{Q}^m$ is the mapping $\chi_{\mathcal{A}}$ assigning $\mathcal{A}$-colors to m-tuples of rationals such that for every $\bar{r} \in \mathbb{Q}^m$ we have $P\bar{\mathsf{a}} \in \chi_{\mathcal{A}}(\bar{r})$ if and only if $\langle \bar{\mathsf{a}}, \bar{r} \rangle \in P^{\mathcal{A}}$.* $\mathcal{A}$-color

Having the coloring $\chi_{\mathcal{A}}$ at hand, it is easy to formulate a uniformity property for any given structure $\mathcal{A}$. Two tuples $\bar{r}, \bar{r}' \in \mathbb{Q}^m$ are treated *uniformly* by $\mathcal{A}$, if the colors $\chi_{\mathcal{A}}(\bar{r})$ and $\chi_{\mathcal{A}}(\bar{r}')$ agree. Put differently, $\mathcal{A}$ does not distinguish $\bar{r}$ from $\bar{r}'$.

Definition 10.2.6 ($\mathcal{J}_{\mathcal{A}}$-uniformity)**.** *A structure $\mathcal{A}$ is $\mathcal{J}_{\mathcal{A}}$-*uniform *if $\chi_{\mathcal{A}}$ colors each and every $\sim_{\mathcal{J}_{\mathcal{A}}}$-equivalence class uniformly, i.e. for all $\sim_{\mathcal{J}_{\mathcal{A}}}$-equivalent tuples $\bar{r}, \bar{r}'$ we have $\chi_{\mathcal{A}}(\bar{r}) = \chi_{\mathcal{A}}(\bar{r}')$.*

We next show that there exists a $\mathcal{J}_{\mathcal{A}}$-uniform model $\mathcal{A}$ of N, if N is satisfiable. Since such a model does not distinguish between $\mathcal{J}_{\mathcal{A}}$-equivalent m-tuples, and as there are only finitely many equivalence classes induced by $\sim_{\mathcal{J}_{\mathcal{A}}}$, only a finite amount of information is required to describe the structure $\mathcal{A}$. This insight will give rise to a decision procedure that nondeterministically guesses how each and every equivalence class shall be treated by the uniform model.

Given some model $\mathcal{A}$ of N, the following lemma assumes the existence of certain finite sets Q_i with a fixed finite cardinality which are subsets of the open intervals in $\mathcal{J}_{\mathcal{A}}$. All $\mathcal{J}_{\mathcal{A}}$-equivalent m-tuples that can be constructed from the rationals belonging to the Q_i are required to be colored identically by $\chi_{\mathcal{A}}$. The existence of the sets Q_i is stipulated (and proved) in Lemma 10.2.9.

Lemma 10.2.7. *Let λ be the maximal number of distinct base-sort variables in any single clause in N. In case of $\lambda < m$ we set $\lambda := m$. Let $\mathcal{A}$ be a model of N. Let $J_0, \{r_1\}, J_1, \ldots, \{r_\kappa\}, J_\kappa$ be an enumeration of all intervals in $\mathcal{J}_{\mathcal{A}}$ sorted in ascending order with the J_i being the open intervals. Suppose we are given a collection of finite sets $Q_0, \ldots, Q_\kappa$ possessing the following properties:* λ J_i, r_i Q_i

(i) $Q_i \subseteq J_i$ and $|Q_i| = \lambda$ for every i.

(ii) Let $Q := \bigcup_i Q_i \cup \{r_1, \ldots, r_\kappa\}$. For all $\mathcal{J}_{\mathcal{A}}$-equivalent m-tuples $\bar{q}, \bar{q}' \in Q^m$ we have $\chi_{\mathcal{A}}(\bar{q}) = \chi_{\mathcal{A}}(\bar{q}')$. Q

Then, we can construct a model $\mathcal{B}$ of N that is $\mathcal{J}_{\mathcal{B}}$-uniform and that interprets the free sort $\mathcal{S}$ with a finite set. Moreover, $\mathcal{B}$ interprets all constant symbols in N in the same way $\mathcal{A}$ does.

Proof.

Claim I: Let μ be any positive integer with $1 \leq \mu \leq \lambda$. For each of the finitely many equivalence classes in $\mathbb{Q}^\mu/_{\sim_{\mathcal{J}_\mathcal{A}}}$ we find a representative lying in Q^μ.

Proof: Given any equivalence class $[\bar{r}]_{\sim_{\mathcal{J}_\mathcal{A}}} \in \mathbb{Q}^\mu/_{\sim_{\mathcal{J}_\mathcal{A}}}$, we define the following ascending sequences for every i, $0 \leq i \leq \kappa$:

$s_{i,1} < \ldots < s_{i,k_i}$ enumerating in ascending order all the values occurring in $\bar{r}$ that stem from J_i, and

$q_{i,1} < \ldots < q_{i,\lambda}$ comprising all rationals from Q_i in ascending order.

In every $Q_i \subseteq J_i$ we find $\lambda \geq \mu \geq k_i$ distinct rationals.

We can now construct a tuple $\bar{q}' \in [\bar{r}]_{\sim_{\mathcal{J}_\mathcal{A}}} \cap Q^\mu$ by setting

$$q'_\ell := \begin{cases} c^\mathcal{A} & \text{if } r_\ell = c^\mathcal{A} \text{ for some } c \in \text{consts}(N) \text{ of sort } \mathbb{Q}, \\ q_{i,j} & \text{if } r_\ell = s_{i,j} \text{ for some } i,\ 0 \leq i \leq \kappa, \text{ and some } j,\ 1 \leq j \leq k_i, \end{cases}$$

for every ℓ, $1 \leq \ell \leq \mu$. Clearly, $\bar{r}$ and $\bar{q}'$ are $\mathcal{J}_\mathcal{A}$-equivalent. ◊

c_i, $\widehat{\mathcal{S}}$, $\mathcal{B}$

Let $c_1, \ldots, c_n$ be an enumeration of all constant symbols of sort $\mathcal{S}$ that occur in N and let $\widehat{\mathcal{S}}$ denote the set $\{\mathsf{a} \in \mathcal{S}^\mathcal{A} \mid \mathsf{a} = c_i^\mathcal{A} \text{ for some } c_i\}$. We construct the structure $\mathcal{B}$ as follows. We set $\mathcal{S}^\mathcal{B} := \widehat{\mathcal{S}}$, and for every constant symbol c occurring in N we set $c^\mathcal{B} := c^\mathcal{A}$. Moreover, for every uninterpreted predicate symbol P occurring in N and for all tuples $\bar{\mathsf{a}} \in \widehat{\mathcal{S}}^{m'}$ and $\bar{r} \in \mathbb{Q}^m$ we pick some tuple $\bar{q} \in Q^m$ which is $\mathcal{J}_\mathcal{A}$-equivalent to $\bar{r}$, and we define $P^\mathcal{B}$ so that

$$\langle \bar{\mathsf{a}}, \bar{r} \rangle \in P^\mathcal{B} \quad \text{if and only if} \quad \langle \bar{\mathsf{a}}, \bar{q} \rangle \in P^\mathcal{A} \ .$$

Claim II: The structure $\mathcal{B}$ is $\mathcal{J}_\mathcal{B}$-uniform.

Proof: By construction of $\mathcal{B}$ and by Requirement (ii). ◊

We next show $\mathcal{B} \models N$. Consider any clause $C = \Lambda \wedge \Gamma \to \Delta$ in N and let β be any variable assignment ranging over $\mathcal{S}^\mathcal{B} \cup \mathbb{Q}$. Starting from β, we derive a special variable assignment γ_C as follows. Let $x_1, \ldots, x_{\lambda_C}$ be an enumeration of all base-sort variables occurring in C. By Claim I, there is some tuple $\langle q_1, \ldots, q_{\lambda_C} \rangle \in Q^{\lambda_C}$ such that $\langle q_1, \ldots, q_{\lambda_C} \rangle \sim_{\mathcal{J}_\mathcal{A}} \langle \beta(x_1), \ldots, \beta(x_{\lambda_C}) \rangle$. We define $\gamma_C(x_i) := q_i$ for every i, $1 \leq i \leq \lambda_C$. For all other base-sort variables, γ_C can be defined arbitrarily. For every free-sort variable u we set $\gamma_C(u) := \beta(u)$. We observe

$$\langle \beta(x_1), \ldots, \beta(x_{\lambda_C}) \rangle \sim_{\mathcal{J}_\mathcal{B}} \langle \gamma_C(x_1), \ldots, \gamma_C(x_{\lambda_C}) \rangle \ . \tag{10.1}$$

As $\mathcal{A}$ is a model of N, we get $\mathcal{A}, \gamma_C \models C$. By case distinction on why $\mathcal{A}, \gamma_C \models C$ holds, we can infer $\mathcal{B}, \beta \models C$ as follows:

Case $\mathcal{A}, \gamma_C \not\models t \lhd t'$ for some LRA atom $t \lhd t'$ in Λ, where t, t' are base-sort variables or ground LRA terms. Notice that, since we assume C to be in BSR(SLR) normal form, if t is a variable, then t' is either a variable or a constant symbol, and if t' is a variable, then t is either a variable or a constant symbol. Since $\mathcal{B}$ and $\mathcal{A}$ interpret constant symbols in the same way and due to (10.1), we conclude $\mathcal{B}, \beta \not\models t \lhd t'$.

Case $\mathcal{A}, \gamma_C \not\models t \approx t'$ for some free-sort equation $t \approx t' \in \Gamma$. In this case, t and t' are either variables or constant symbols of the free sort, which means they do not contain subterms of the base sort. Since $\mathcal{B}$ and $\mathcal{A}$ behave identical on free-sort constant symbols and $\beta(u) = \gamma_C(u)$ for every variable $u \in V_\mathcal{S}$, we have $\mathcal{B}, \beta \not\models t \approx t'$.

Case $\mathcal{A}, \gamma_C \models t \approx t'$ for some $t \approx t' \in \Delta$. In analogy to the above case, we get $\mathcal{B}, \beta \models t \approx t'$.

Case $\mathcal{A}, \gamma_C \not\models P(t'_1, \ldots, t'_{m'}, t_1, \ldots, t_m)$ for some non-equational atom $P(t'_1, \ldots, t'_{m'}, t_1, \ldots, t_m)$ in Γ. This translates to

$$\langle \mathcal{A}(\gamma_C)(t'_1), \ldots, \mathcal{A}(\gamma_C)(t'_{m'}), \mathcal{A}(\gamma_C)(t_1), \ldots, \mathcal{A}(\gamma_C)(t_m) \rangle \notin P^{\mathcal{A}}.$$

Moreover, since N is in BSR(SLR) normal form,, C must belong to N_{BSR} and thus each t_j is either a variable of sort $\mathbb{Q}$ or a Skolem constant of sort $\mathbb{Q}$. By definition of γ_C, we have $\mathcal{A}(\gamma_C)(t_j) \in Q$ for every j, $1 \leq j \leq m$. Therefore, and by construction of $\mathcal{B}$,

$$\langle \mathcal{A}(\gamma_C)(t'_1), \ldots, \mathcal{A}(\gamma_C)(t'_{m'}), \mathcal{A}(\gamma_C)(t_1), \ldots, \mathcal{A}(\gamma_C)(t_m) \rangle \notin P^{\mathcal{B}}.$$

We observe the following properties:

(I) We have $\mathcal{A}(\gamma_C)(t'_j) = \mathcal{B}(\beta)(t'_j)$ for every j, $1 \leq j \leq m'$, due to the definition of $\mathcal{B}$ and γ_C.

(II) Since $\mathcal{A}$ and $\mathcal{B}$ interpret constant symbols in the same way, we get $\mathcal{A}(\gamma_C)(t_j) = \mathcal{B}(\gamma_C)(t_j)$ for every j, $1 \leq j \leq m$.

(III) The definition of γ_C entails that $\langle \mathcal{B}(\gamma_C)(t_1), \ldots, \mathcal{B}(\gamma_C)(t_m) \rangle$ and $\langle \mathcal{B}(\beta)(t_1), \ldots, \mathcal{B}(\beta)(t_m) \rangle$ are $\mathcal{J}_{\mathcal{B}}$-equivalent.

The first two observations imply

$$\langle \mathcal{B}(\beta)(t'_1), \ldots, \mathcal{B}(\beta)(t'_{m'}), \mathcal{B}(\gamma_C)(t_1), \ldots, \mathcal{B}(\gamma_C)(t_m) \rangle \notin P^{\mathcal{B}}.$$

Due to this result and the fact that $\mathcal{B}$ is $\mathcal{J}_{\mathcal{B}}$-uniform (Claim II), the third observation leads to $\langle \mathcal{B}(\beta)(t'_1), \ldots, \mathcal{B}(\beta)(t'_{m'}), \mathcal{B}(\beta)(t_1), \ldots, \mathcal{B}(\beta)(t_m) \rangle \notin P^{\mathcal{B}}$.

Put differently, we have $\mathcal{B}, \beta \not\models P(t'_1, \ldots, t'_{m'}, t_1, \ldots, t_m)$.

Case $\mathcal{A}, \gamma_C \models P(t'_1, \ldots, t'_{m'}, t_1, \ldots, t_m)$ for some non-equational atom $P(t'_1, \ldots, t'_{m'}, t_1, \ldots, t_m)$ in Δ. In analogy to the previous case we may infer $\mathcal{B}, \beta \models P(t'_1, \ldots, t'_{m'}, t_1, \ldots, t_m)$.

Altogether, we have shown $\mathcal{B} \models N$. □

In order to show that uniform models always exist for satisfiable clause sets N, we still need to prove the existence of the sets Q_i required in Lemma 10.2.7. We use Lemma 10.1.4 to show this. As an auxiliary result, we first show a correspondence between the equivalence classes with respect to $\sim_{\mathcal{J}_{\mathcal{A}}}$ and mappings $\rho : [m] \to [|\mathcal{J}_{\mathcal{A}}|] \times [m]$.

Lemma 10.2.8. *Let $\mathcal{A}$ be any structure. Let $J_1, \{q_1\}, J_2, \{q_2\}, \ldots, \{q_\kappa\}, J_{\kappa+1}$ be an enumeration of all intervals in $\mathcal{J}_{\mathcal{A}}$ sorted in ascending order with the J_i being the open intervals. Let $S \in \mathbb{Q}^m/_{\sim_{\mathcal{J}_{\mathcal{A}}}}$ be any equivalence class with respect to $\sim_{\mathcal{J}_{\mathcal{A}}}$ containing m-tuples. There is some mapping $\rho : [m] \to [|\mathcal{J}_{\mathcal{A}}|] \times [m]$ such that* J_i, q_i, ρ

(i) whenever $\rho(i) = \langle k, \ell \rangle$ with $k > \kappa + 1$ then $\ell = 1$, and

(ii) for all ascending tuples

$$\begin{aligned}
\bar{r}_1 &= \langle r_{\langle 1,1 \rangle}, \ldots, r_{\langle 1,m \rangle} \rangle \in J_1^m, \\
&\vdots \\
\bar{r}_{\kappa+1} &= \langle r_{\langle \kappa+1,1 \rangle}, \ldots, r_{\langle \kappa+1,m \rangle} \rangle \in J_{\kappa+1}^m, \\
\bar{r}_{\kappa+2} &= \langle r_{\langle \kappa+2,1 \rangle} \rangle = \langle q_1 \rangle \\
&\vdots \\
\bar{r}_{2\kappa+1} &= \langle r_{\langle 2\kappa+1,1 \rangle} \rangle = \langle q_\kappa \rangle
\end{aligned}$$

we have $\langle r_{\rho(1)}, \ldots, r_{\rho(m)} \rangle \in S$, and

(iii) for every tuple $\langle s_1, \ldots, s_m \rangle \in S$ there exist ascending tuples $\bar{r}_1, \ldots, \bar{r}_{2\kappa+1}$ defined like in (ii) such that $\langle s_1, \ldots, s_m \rangle = \langle r_{\rho(1)}, \ldots, r_{\rho(m)} \rangle$.

Proof. Fix any $S \in \mathbb{Q}^m/_{\sim_{\mathcal{J}_\mathcal{A}}}$. Let $\bar{s}'$ be some representative taken from S, i.e. $S = [\bar{s}']_{\sim_{\mathcal{J}_\mathcal{A}}}$. Given $\bar{s}'$, we construct $2\kappa+1$ possibly empty sequences $\bar{s}''_k := \langle s''_{k,1}, s''_{k,2}, \ldots\rangle$, such that every $\bar{s}''_k$ with $k \leq \kappa+1$ lists all elements of $\bar{s}'$ in ascending order that lie in J_k, and every $\bar{s}''_k$ with $k > \kappa+1$ contains exactly the value $q_{k-\kappa-1}$. We construct the mapping ρ in such a way that $\rho(i) = \langle k, \ell\rangle$ holds if and only if $s'_i = s''_{k,\ell}$.

Now let $\bar{r}_1, \ldots, \bar{r}_{2\kappa+1}$ be any tuples of rationals chosen in accordance with requirement (ii). It is easy to verify that $\bar{r}_\rho := \langle r_{\rho(1)}, \ldots, r_{\rho(m)}\rangle$ is $\mathcal{J}_\mathcal{A}$-equivalent to $\bar{s}'$, i.e. $\bar{r}_\rho$ belongs to S.

In order to show (iii), we construct the tuples $\bar{r}_1, \ldots, \bar{r}_{2\kappa+1}$ from $\langle s_1, \ldots, s_m\rangle$ in the same way we have constructed the $\bar{s}''_k$ from $\bar{s}'$ above. In addition, we pad them with suitable values from the respective intervals J_k to reach the length m for every tuple. Then, it is easy to verify that $\langle s_1, \ldots, s_m\rangle = \langle r_{\rho(1)}, \ldots, r_{\rho(m)}\rangle$. □

J_i, r_i

Lemma 10.2.9. *Let $\mathcal{A}$ be any structure. Let $J_0, \{r_1\}, J_1, \ldots, \{r_\kappa\}, J_\kappa$ be an enumeration of all intervals in $\mathcal{J}_\mathcal{A}$ sorted in ascending order with the J_i being the open intervals. Let λ be any positive integer. There is a collection of finite sets $Q_0, \ldots, Q_\kappa$ such that Requirements (i) and (ii) of Lemma 10.2.7 are met:*

(i) We have $Q_i \subseteq J_i$ and $|Q_i| = \lambda$ for every i, $1 \leq i \leq \kappa+1$.

(ii) Let $Q := \bigcup_i Q_i \cup \{r_1, \ldots, r_\kappa\}$. For all $\mathcal{J}_\mathcal{A}$-equivalent m-tuples $\bar{q}, \bar{q}' \in Q^m$ we have $\chi_\mathcal{A}(\bar{q}) = \chi_\mathcal{A}(\bar{q}')$.

Proof. Let the sets $Q_1, \ldots, Q_{\kappa+1}$ be the $Q_1, \ldots, Q_p$ which we obtain by virtue of Lemma 10.1.4 when we set $n := \lambda$, $p := \kappa+1$, $\chi := \chi_\mathcal{A}$, $R_1 := J_1, \ldots, R_p := J_{\kappa+1}$. Requirement (i) is obviously satisfied for $Q_1, \ldots, Q_{\kappa+1}$. By Lemma 10.2.8, the equivalence class to which any two given $\mathcal{J}_\mathcal{A}$-equivalent tuples $\bar{q}, \bar{q}'$ belong corresponds to some mapping $\rho : [m] \to [2\kappa+1] \times [m]$. Part (ii) of Lemma 10.2.8 states that $\bar{q}$ can be written in the form $\langle r_{\rho(1)}, \ldots, r_{\rho(m)}\rangle$ for appropriate values $r_{\langle k,\ell\rangle}$ and $\bar{q}'$ can be represented in the form $\langle r'_{\rho(1)}, \ldots, r'_{\rho(m)}\rangle$ for appropriate $r'_{\langle k,\ell\rangle}$. We then know by Lemma 10.1.4 that $\chi_\mathcal{A}(\bar{q}) = \chi_\mathcal{A}(\langle r_{\rho(1)}, \ldots, r_{\rho(m)}\rangle) = \chi_\mathcal{A}(\langle r'_{\rho(1)}, \ldots, r'_{\rho(m)}\rangle) = \chi_\mathcal{A}(\bar{q}')$. □

Lemmas 10.2.7 and 10.2.9 together entail the existence of some $\mathcal{J}_\mathcal{A}$-uniform model $\mathcal{A} \models N$ with a finite free-sort domain $\mathcal{S}^\mathcal{A}$, if N is satisfiable.

Corollary 10.2.10. *If N has a model, then it has a model $\mathcal{A}$ that is $\mathcal{J}_\mathcal{A}$-uniform and that interprets the sort $\mathcal{S}$ with some finite set.*

Corollary 10.2.10 provides the key tool for devising a decision procedure for finite BSR(SLR) clause sets. But before we present such a procedure, we need to develop a variant of Lemma 10.2.7 that is easier to handle from the computational point of view.[1] Recall that we assume N to be in normal form (cf. Definition 10.0.3). Therefore, N can be partitioned into $N_\mathbb{Q}$, N_{BSR}, where $N_{\text{BSR}} \subseteq N$ is a subset that contains exactly the clauses $\Lambda \wedge \Gamma \to \Delta$ from N with nonempty Γ or Δ. By Requirement (b) of Definition 10.0.3, we may assume that N_{BSR} does not contain any symbol from the arithmetic part of the underlying vocabulary, except for $<, \leq$. That is, N_{BSR} does neither contain any rational numbers nor any arithmetic operators. Our aim is to treat $N_\mathbb{Q}$ and N_{BSR} in isolation: combining a decision procedure for LRA and one for BSR sentences over uninterpreted vocabularies.

c_i, d_i

Let $c_1, \ldots, c_k$ be an enumeration of all the constant symbols in $N_\mathbb{Q} \cup N_{\text{BSR}}$ that are of the sort $\mathbb{Q}$ and let $d_1, \ldots, d_\ell$ be an enumeration of all free-sort constant symbols occurring in N_{BSR}. We

[1] The author of the present thesis is indebted to Pascal Fontaine for pointing out in a discussion in November 2017 that Lemma 10.2.7 and Corollary 10.2.10 alone are not sufficient for devising a decision procedure for BSR(SLR).

define ψ to be the following BSR sentence (with constant symbols)

$$\begin{aligned}\psi := {} & \big(\forall v.\, \mathrm{Rat}(v) \leftrightarrow \neg\mathrm{Free}(v)\big) \wedge \big(\forall xy.\, x < y \vee x \leq y \rightarrow \mathrm{Rat}(x) \wedge \mathrm{Rat}(y)\big) \\ & \wedge \bigwedge_{1\leq i\leq k} \mathrm{Rat}(c_i) \wedge \bigwedge_{1\leq j\leq \ell} \mathrm{Free}(d_j) \\ & \wedge \Big(\forall xyz.\, \mathrm{Rat}(x) \wedge \mathrm{Rat}(y) \wedge \mathrm{Rat}(z) \longrightarrow \Big(\big(\neg x < x\big) \\ & \qquad\qquad \wedge \big(x < y \wedge y < z \rightarrow x < z\big) \\ & \qquad\qquad \wedge \big(x \approx y \vee x < y \vee y < x\big) \\ & \qquad\qquad \wedge \big(x \leq y \leftrightarrow x \approx y \vee x < y\big)\Big)\Big)\,,\end{aligned} \qquad \psi$$

which explicitly stipulates sort membership for constant symbols and contains the axioms of total orders for $<$ and $\leq$. We assume, without loss of generality, that the predicate symbols Rat and Free do not occur in N. The length of ψ is linear in the length of $N_{\mathbb{Q}} \cup N_{\mathrm{BSR}}$. Notice that we use the equality sign $\approx$ instead of $=$, and thus refrain from treating arithmetic variables in a privileged way. The reason will become apparent in Lemma 10.2.11.

Let $\bar{\mathrm{u}}$ be some tuple listing all variables from $\mathrm{vars}(N_{\mathrm{BSR}}) \cap \mathrm{Var}_{\mathcal{S}}$ and let $\bar{\mathrm{x}}$ be some tuple listing all variables from $\mathrm{vars}(N_{\mathrm{BSR}}) \cap \mathrm{Var}_{\mathbb{Q}}$. We define the sentence $\varphi_{N_{\mathrm{BSR}}}$

$$\varphi_{N_{\mathrm{BSR}}} := \psi \wedge \forall \bar{\mathrm{u}}\bar{\mathrm{x}}.\, \Big(\bigwedge_{u\in\bar{\mathrm{u}}} \mathrm{Free}(u) \wedge \bigwedge_{x\in\bar{\mathrm{x}}} \mathrm{Rat}(x)\Big) \rightarrow \bigwedge_{C(\bar{\mathrm{u}},\bar{\mathrm{x}})\in N_{\mathrm{BSR}}} C(\bar{\mathrm{u}},\bar{\mathrm{x}})\ ,$$

which is evidently equivalent to some BSR sentence and whose length is linear in the length of $N_{\mathbb{Q}} \cup N_{\mathrm{BSR}}$. In addition, we define the sentence $\eta_{\preceq}$ for any total *preorder* $\preceq$ (a reflexive and transitive binary relation) over the constant symbols $c_1, \ldots, c_k$ as follows. We write $c_i \prec c_j$ if and only if we have $c_i \preceq c_j$ and $c_j \not\preceq c_i$. Suppose that $c_{j_1} \prec \ldots \prec c_{j_{k'}}$ is a maximal $\prec$-chain with $k' \leq k$. Let λ be the maximal number of distinct base-sort variables in any single clause in N_{BSR}. In case of $\lambda < m$ we set $\lambda := m$. Let $\bar{\mathrm{z}}_0, \ldots, \bar{\mathrm{z}}_{k'}$ be pairwise-disjoint tuples of first-order variables of length λ each. We set $\preceq$, $c_i \prec c_j$, λ, $\bar{\mathrm{z}}_i$

$$\begin{aligned}\eta_{\preceq} := \exists \bar{\mathrm{z}}_0 \ldots \bar{\mathrm{z}}_{k'}.\, & \Big(\bigwedge_{1\leq i\leq \lambda-1} z_{0,i} < z_{0,i+1}\Big) \\ & \wedge\ z_{0,\lambda} < c_{j_1}\ \wedge\ c_{j_1} < z_{1,1} \\ & \wedge\ \Big(\bigwedge_{1\leq i\leq \lambda-1} z_{1,i} < z_{1,i+1}\Big) \\ & \wedge\ z_{1,\lambda} < c_{j_2}\ \wedge\ c_{j_1} < z_{2,1} \\ & \ \ \vdots \\ & \wedge\ z_{k'-1,\lambda} < c_{j_{k'}}\ \wedge\ c_{j_{k'}} < z_{k',1} \\ & \wedge\ \Big(\bigwedge_{1\leq i\leq \lambda-1} z_{k,i} < z_{k,i+1}\Big) \\ & \wedge \bigwedge_{\substack{c_i\preceq c_j \\ \wedge\, c_j\preceq c_i}} c_i \approx c_j\ .\end{aligned} \qquad \eta_{\preceq}$$

Written in a more convenient notation, $\eta_{\preceq}$ establishes the chain

$$\begin{aligned}z_{0,1} < \ldots < z_{0,\lambda} < c_{j_1} < z_{1,1} < \ldots < z_{1,\lambda} < c_{j_2} < \ \ldots \\ < c_{j_{k'-1}} < z_{k'-1,1} < \ldots < z_{k'-1,\lambda} < c_{j_{k'}} < z_{k',1} < \ldots < z_{k',\lambda}\end{aligned}$$

and identifies c_i and c_j whenever $c_i \preceq c_j \preceq c_i$. Notice that the length of $\eta_{\preceq}$ is at most quadratic in the length of $N_{\mathbb{Q}} \cup N_{\mathrm{BSR}}$.

For the following variant of Lemma 10.2.7 we make an exception for the sentences $\varphi_{N_{\mathrm{BSR}}}$ and $\eta_{\preceq}$ and treat $<, \leq$ as uninterpreted predicate symbols and consider both sentences without sorts.

Lemma 10.2.11. *Let $\preceq$ be any total preorder over the constant symbols $c_1, \ldots, c_k$. Suppose there is a model $\mathcal{A} \models \varphi_{N_{\mathrm{BSR}}} \wedge \eta_{\preceq}$ with a single-sorted domain and in which $<, \leq$ are treated as uninterpreted predicate symbols. Assume that $\mathcal{A}$'s domain is minimal, i.e. $\mathcal{A}$ does not contain any substructure that also satisfies $\varphi_{N_{\mathrm{BSR}}} \wedge \eta_{\preceq}$ — notice that this entails that A is finite. Furthermore, assume that for all m-tuples $\bar{q}, \bar{q}'$ of elements from $\mathrm{Rat}^{\mathcal{A}}$ which are $\mathcal{J}_{\mathcal{A}}$-equivalent[2] we have $\chi_{\mathcal{A}}(\bar{q}) = \chi_{\mathcal{A}}(\bar{q}')$.*

Then, we can construct a model $\mathcal{B}$ of N_{BSR} that is $\mathcal{J}_{\mathcal{B}}$-uniform, contains the rational numbers as subdomain, interprets the predicate symbols $<, \leq$ as the usual relations over the rationals, and interprets the free sort $\mathcal{S}$ with some finite set. Moreover, we have $\mathcal{B} \models c_i < c_j$ if and only if $\mathcal{A} \models c_i < c_j$ if and only if $c_i \prec c_j$.

The proof of Lemma 10.2.11 proceeds along the same lines as the proof of Lemma 10.2.7 does. We only need to switch from the setting of N_{BSR} mixing interpreted arithmetic relations over the rationals with uninterpreted predicate symbols to the point of view of the purely uninterpreted setting of $\varphi_{N_{\mathrm{BSR}}} \wedge \eta_{\preceq}$. Moreover, Corollary 10.2.10 guarantees the existence of a model $\mathcal{A}$ as described in Lemma 10.2.11 whenever N_{BSR} is satisfiable (in the arithmetic setting).

Proposition 10.2.12. *If N has a model $\mathcal{B}$, then there is some total preorder $\preceq$ over the base-sort Skolem constant symbols $c_1, \ldots, c_k$ occurring in N and a $\mathcal{J}_{\mathcal{A}}$-uniform[3] model $\mathcal{A} \models \varphi_{N_{\mathrm{BSR}}} \wedge \eta_{\preceq}$ with a finite domain. Moreover, we have $\mathcal{B} \models c_i < c_j$ if and only if $\mathcal{A} \models c_i < c_j$ if and only if $c_i \prec c_j$.*

Now we have all pieces together to devise a nondeterministic decision procedure for finite BSR(SLR) clause sets in normal form. Consider such a clause set N. Recall that since N is in normal form, we can divide N into two disjoint parts $N_{\mathbb{Q}}$ and N_{BSR} such that N_{BSR} does neither contain any rational numbers nor any arithmetic operators (except for the predicate symbols $<, \leq$). Our previous observations lead to the following nondeterministic decision procedure:

(I) Nondeterministically fix a total preorder $\preceq$ over the set of all base-sort Skolem constants occurring in $N_{\mathbb{Q}} \cup N_{\mathrm{BSR}}$.

Define a clause set $N_{\preceq}$ that enforces $\preceq$ for base-sort Skolem constants:

$N_{\preceq}$

$$N_{\preceq} := \left\{ c > c' \to \mathtt{false} \;\middle|\; c \preceq c' \right\} .$$

(II) Check whether the clause set $N_{\mathbb{Q}} \cup N_{\preceq}$ is satisfiable under $\mathbb{Q}$, that is, check whether there is some assignment $\gamma : \{c_1, \ldots, c_k\} \to \mathbb{Q}$ such that $\mathbb{Q} \models \big(N_{\mathbb{Q}} \cup N_{\preceq}\big)\big[c_1/\gamma(c_1), \ldots, c_k/\gamma(c_k)\big]$ where $\big(N_{\mathbb{Q}} \cup N_{\preceq}\big)\big[c_1/\gamma(c_1), \ldots, c_k/\gamma(c_k)\big]$ denotes the syntactic replacement of every c_i with $\gamma(c_i)$ in $N_{\mathbb{Q}} \cup N_{\preceq}$.

(III) Check whether the (single-sorted) BSR sentence $\varphi_{N_{\mathrm{BSR}}} \wedge \eta_{\preceq}$ — transformed into prenex normal form with a $\exists^*\forall^*$ quantifier prefix — is satisfied by some model $\mathcal{A}$ that is $\mathcal{J}_{\mathcal{A}}$-uniform.

(IV) If both Step (II) and Step (III) succeed, then N is satisfiable.

Step (II) relies on the fact that $N_{\mathbb{Q}} \cup N_{\preceq}$ is variable free and that the existential fragment of linear rational arithmetic is decidable (cf. Proposition 10.2.13). Notice also that Steps (II) and (III) could be done in any order. By Lemma 10.2.11 and Proposition 10.2.12, the procedure is a correct and complete decision procedure for the satisfiability of finite BSR(SLR) clause sets in normal form.

Next, we investigate the time complexity of the outlined decision procedure. To this end, we first argue that Step (II) can be done in nondeterministic polynomial time.

[2] Although $\mathcal{J}_{\mathcal{A}}$-equivalence and the coloring function $\chi_{\mathcal{A}}$ are technically defined for a different setting, we reuse the definitions in Lemma 10.2.11 and in Proposition 10.2.12 with their intended meaning without formally adapting them to the new setting.

[3] See Footnote 2.

Proposition 10.2.13. *Let* $\varphi_{\mathbb{Q}} := \exists \bar{\mathrm{v}}.\ \bigwedge_{C \in N_{\mathbb{Q}} \cup N_{\preceq}} C\big[c_1/v_1, \ldots, c_k/v_k\big]$ *where* $\bar{\mathrm{v}}$ *contains* k *fresh variables* $v_1, \ldots, v_k$ *of sort* $\mathbb{Q}$ *and* $C\big[c_1/v_1, \ldots, c_k/v_k\big]$ *denotes the clause* C *after replacing every* c_i *in* C *with* v_i*. The question whether* $\mathbb{Q} \models \varphi_{\mathbb{Q}}$ *holds can be decided nondeterministically in polynomial time with respect to* $\|N_{\mathbb{Q}} \cup N_{\preceq}\|$*, i.e. with respect to the length of the binary encoding of the clause set* $N_{\mathbb{Q}} \cup N_{\preceq}$*.*

Proof sketch. We devise the following nondeterministic decision procedure:

(1) Let At be the set of all atoms occurring in $\varphi_{\mathbb{Q}}$. Nondeterministically choose a subset $S \subseteq \mathrm{At}$.

(2) Construct the propositional counterpart φ_{prop} of $\varphi_{\mathbb{Q}}$ by replacing every arithmetic atom A in $\varphi_{\mathbb{Q}}$ with the propositional variable p_A. Check whether $\big(\bigwedge_{A \in S} p_A\big) \wedge \big(\bigwedge_{A \in \mathrm{At} \setminus S} \neg p_A\big) \rightarrow \varphi_{\mathrm{prop}}$ is a valid propositional formula.

(3) Let $\psi(\bar{\mathrm{v}}) := \big(\bigwedge_{A(\bar{\mathrm{v}}) \in S} A(\bar{\mathrm{v}})\big) \wedge \big(\bigwedge_{A(\bar{\mathrm{v}}) \in \mathrm{At} \setminus S} \neg A(\bar{\mathrm{v}})\big)$ and check whether we have $\mathbb{Q} \models \exists \bar{\mathrm{v}}.\, \psi(\bar{\mathrm{v}})$.

(4) If both Steps (2) and (3) succeed, then $\varphi_{\mathbb{Q}}$ is satisfied under $\mathbb{Q}$.

As the formula $\big(\bigwedge_{A \in S} p_A\big) \wedge \big(\bigwedge_{A \in \mathrm{At} \setminus S} \neg p_A\big)$ in Step (2) describes a complete assignment of all the propositional variables occurring in φ_{prop}, Step (2) amounts to checking whether φ_{prop} is satisfied under this assignment. Hence, Step (2) can be done in polynomial time (with respect to $\mathrm{len}(\varphi_{\mathrm{prop}})$). Regarding Step (3), conjunctions of linear inequalities with existentially quantified variables over the rational numbers can be solved deterministically in polynomial time (with respect to the length of their binary encoding) via a transformation into a *linear program* (see, for instance, [BM07], pages 217–218 in Section 8.3 and Theorem 8.17). It is well known that for any *feasible* linear program over the rational numbers a solution can be computed in polynomial time [Kha80, GL81, Kar84] (see also [RESW14] and [Sch99], Sections 10, 13, and 14). □

By virtue of Proposition 10.2.13, Step (II) can be done nondeterministically in polynomial time with respect to $\|N_{\mathbb{Q}} \cup N_{\preceq}\|$, where the formula length of $N_{\preceq}$ (and also $\|N_{\preceq}\|$) is at most quadratic in the length of $N_{\mathbb{Q}} \cup N_{\mathrm{BSR}}$. On the other hand, the sentence $\varphi_{N_{\mathrm{BSR}}} \wedge \eta_{\preceq}$ has a formula length that is polynomial in the length of $N_{\mathbb{Q}} \cup N_{\mathrm{BSR}}$. By Proposition 3.1.6, we know that, if $\varphi_{N_{\mathrm{BSR}}} \wedge \eta_{\preceq}$ is satisfiable, then it has a model whose domain contains at most as many elements as $\varphi_{N_{\mathrm{BSR}}} \wedge \eta_{\preceq}$ contains existentially quantified variables plus constant symbols.

Unfortunately, we cannot invoke Proposition 5.0.1 directly to obtain an upper bound regarding the computational complexity of Step (III), because the proposition only speaks about the general first-order satisfiability problem and Step (III) is about satisfiability with respect to a restricted class of structures. However, the decision procedure underlying Proposition 5.0.1 (see, e.g., Proposition 6.0.4 in [BGG97]) starts with nondeterministically guessing a structure $\mathcal{A}$ that is a candidate model for the sentence at hand, and then checks in deterministic exponential time whether $\mathcal{A}$ is indeed a model. It is easy to see that we could restrict the guessing step to structures that are $\mathcal{J}_{\mathcal{A}}$-uniform[4]. The latter in fact amounts to an additional step for checking whether the candidate model is indeed $\mathcal{J}_{\mathcal{A}}$-uniform, which can be done deterministically in time that is exponential in the length of the considered sentence. Hence, by a modified variant of Proposition 5.0.1, we conclude that the satisfiability problem that needs to be solved in Step (III) belongs to NExpTime, and, as the problem is at least as hard as BSR-Sat, it is in fact NExpTime-complete.

Theorem 10.2.14. *The satisfiability problem for finite BSR(SLR) clause sets is decidable, and for clause sets in BSR(SLR) normal form the problem is* NExpTime*-complete.*

By Lemma 10.0.7, every finite BSR(SLR) clause set N can be transformed into an equisatisfiable clause set N' in BSR(SLR) normal form. The occurring blowup is such that (a) the length of N' is at most exponential in the length of N, (b) the number of variables occurring in any clause in N' is not larger than the number of variables occurring in any clause in N, (c) the number of uninterpreted constant symbols occurring in N' is linear in the length of N. Hence, by

[4] See Footnote 2 on page 240.

Proposition 10.2.13 plus the modified variant of Proposition 5.0.1 (see above), Steps (I) to (IV) can still be done nondeterministically in time that is at most exponential in the length of the original clause set N.

Corollary 10.2.15. *The satisfiability problem for finite BSR(SLR) clause sets is* NExpTime*-complete.*

10.3 Shifting Perspective: BSR(SLR) from the Viewpoint of Combinations of Theories

In automated reasoning there are often specialized decision procedures tailored towards specific theories, such as (fragments of) arithmetic over the rationals or integers, equality over uninterpreted function symbols, and theories concerning data structures such as arrays, bit vectors, pointers, and strings. In applications originating from verification of software systems, for instance, it is often necessary to be able to reason about formulas that are based on more than one of these theories. In such applications, one mostly considers existentially quantified formulas and universal quantification is not allowed. Indeed, such syntactic restrictions are one possible way to make certain satisfiability problems decidable which would be undecidable otherwise. From an engineering perspective it is then desirable to combine several decision procedures, each capable of reasoning in one component theory, into a procedure that can handle the combined theory. It turns out that this is a non-trivial task, which can, however, be solved in a satisfying way in certain practically relevant cases. Two basic approaches for solving this challenge were presented by Nelson and Oppen [NO79, Opp80, Nel84, TH96] and by Shostak [Sho84, RS01, Gan02]. Over the years, combination of theories has received quite a bit of attention. The survey article [MZ02] summarizes the development of the field until 2002. Since then the field has developed further, of course. Here we can only give an incomplete list of works that is intended to mention many of the recent contributors: [SR02, TR03, TZ05, RRT04, RRZ05, CK06, GNZ08, ABRS09, WPK09, TRRK10, Sof13, CFR15, GG18]. A more detailed overview and further references can be found in [BM07], Chapter 10, [KS16], Chapter 10, and [BT18], Section 11.5. A conceptual continuation of the combination-of-theories paradigm in the full first-order setting is embodied in *hierarchic superposition* [BGW92, BGW94, BW13b, BW13a, KW12, Kru13]. The latter is a calculus for reasoning about *first-order logic modulo background theories*, which aims at a tight integration of first-order theorem proving with solvers for the considered background theories. For further references, see [Kru13], in particular the related-works section in Chapter 1, and also [GHW03, Sof14], for instance.

The method by Nelson and Oppen was originally designed for combining theories whose vocabularies are disjoint and which are *stably infinite* (see below). Often, it is attempted to relax the mentioned restrictions towards the to-be-combined theories (disjointness of vocabularies and stable infiniteness). The general setup of the framework is the following. We consider a first-order sentence $\exists \bar{\mathrm{v}}.\, \psi_1(\bar{\mathrm{v}}) \wedge \psi_2(\bar{\mathrm{v}})$ where ψ_1 and ψ_2 are formulas over disjoint vocabularies Σ_1, Σ_2 that may contain uninterpreted or interpreted function and predicate symbols. Hence, the only non-logical symbols common to ψ_1 and ψ_2 are the equality sign $\approx$ and the variables from $\bar{\mathrm{v}}$. In contrast to the rest of the chapter, the interpreted part of the vocabularies Σ_1, Σ_2 is not limited to arithmetic. Usually, ψ_1 and ψ_2 are restricted to existential formulas, i.e. they are required to be equivalent to some $\exists^*$ prefix formula. Then, the only means of exchanging information between the parts ψ_1 and ψ_2 is, in essence, equations $v \approx v'$ and disequations $v \not\approx v'$ with $v, v' \in \bar{\mathrm{v}}$. Given decision procedures $\mathrm{DP}_1, \mathrm{DP}_2$ that can decide satisfiability of $\exists \bar{\mathrm{v}}.\, \psi_1(\bar{\mathrm{v}}) \wedge \eta(\bar{\mathrm{v}})$ and of $\exists \bar{\mathrm{v}}.\, \psi_2(\bar{\mathrm{v}}) \wedge \eta(\bar{\mathrm{v}})$, respectively, where η is any conjunction of (dis)equations $[\neg] v \approx v'$, the Nelson–Oppen approach combines the two into one decision procedure for $\exists \bar{\mathrm{v}}.\, \psi_1(\bar{\mathrm{v}}) \wedge \psi_2(\bar{\mathrm{v}})$ as follows.

DP_1, DP_2

$\sim$

$\eta_\sim(\bar{\mathrm{v}})$

(1) Nondeterministically construct an equivalence relation $\sim$ over the variables in $\bar{\mathrm{v}}$. Let $\eta_\sim$ be the formula

$$\eta_\sim(\bar{\mathrm{v}}) := \Big(\bigwedge_{v \sim v'} v \approx v' \Big) \wedge \Big(\bigwedge_{v \not\sim v'} v \not\approx v' \Big) .$$

(2) Use DP_1 to check whether $\exists \bar{\mathrm{v}}.\, \psi_1(\bar{\mathrm{v}}) \wedge \eta_\sim(\bar{\mathrm{v}})$ is satisfiable with respect to the fixed semantics of the interpreted parts of Σ_1.

(3) Use DP_2 to check whether $\exists \bar{\mathrm{v}}.\, \psi_2(\bar{\mathrm{v}}) \wedge \eta_\sim(\bar{\mathrm{v}})$ is satisfiable with respect to the fixed semantics of the interpreted parts of Σ_2.

(4) If both Steps 2 and Step 3 succeed, then $\exists \bar{\mathrm{v}}.\, \psi_1(\bar{\mathrm{v}}) \wedge \psi_2(\bar{\mathrm{v}})$ is satisfiable with respect to the fixed semantics of the interpreted parts of $\Sigma_1 \cup \Sigma_2$.

The outlined decision procedure is correct and complete provided that the semantic restrictions underlying Σ_1, Σ_2 are such that whenever $\exists \bar{\mathrm{v}}.\, \psi_1(\bar{\mathrm{v}}) \wedge \eta_\sim(\bar{\mathrm{v}})$ and $\exists \bar{\mathrm{v}}.\, \psi_2(\bar{\mathrm{v}}) \wedge \eta_\sim(\bar{\mathrm{v}})$ are satisfiable, then there are (single-sorted) models $\mathcal{A}_1 \models \exists \bar{\mathrm{v}}.\, \psi_1(\bar{\mathrm{v}}) \wedge \eta_\sim(\bar{\mathrm{v}})$ and $\mathcal{A}_2 \models \exists \bar{\mathrm{v}}.\, \psi_2(\bar{\mathrm{v}}) \wedge \eta_\sim(\bar{\mathrm{v}})$ that have infinite domains. This property is referred to as *stable infiniteness*. *stable infiniteness*

The decision procedure that we have developed for finite BSR(SLR) clause sets shows great similarities with the Nelson–Oppen approach to combining decision procedures. In what follows we shall abstract from the BSR(SLR) syntax (in normal form) to a certain extent and generalize the results we have obtained to far.

Consider a sentence $\varphi := \exists \bar{\mathrm{v}}.\, \varphi_1(\bar{\mathrm{v}}) \wedge \exists \bar{\mathrm{y}} \forall \bar{\mathrm{z}}.\, \varphi_2(\bar{\mathrm{v}}, \bar{\mathrm{y}}, \bar{\mathrm{z}})$ with the following properties. The conjunct φ_1 is a formula over the language of rational arithmetic (without uninterpreted constant symbols). The conjunct φ_2 is a relational quantifier-free formula that does neither contain rational numbers nor any arithmetic operators except for $<, \leq$. However, φ_2 may contain uninterpreted predicate symbols. Notice that the BSR(SLR) normal form falls into the syntactic category of φ. The syntax of φ emphasizes the loose connection that φ establishes between the arithmetic part and the non-arithmetic predicates via the order relations $<, \leq$. While in the classical Nelson–Oppen setting the only information the two procedures $\mathrm{DP}_1, \mathrm{DP}_2$ need to exchange are equations $v \approx v'$ over existentially quantified variables v, v', in the case considered here, component decision procedures need to exchange information about the order relations $<, \leq$ that are common to the two parts φ_1 and φ_2. More precisely, only the relative positions of the shared variables from $\bar{\mathrm{v}}$ with respect to $<$ need to be exchanged. This leads to the following adapted decision procedure, which is based on the one for finite BSR(SLR) clause sets from Section 10.2: $\varphi,\ \varphi_1,\ \varphi_2$

(1) Nondeterministically choose a total preorder $\preceq$ over the set of all variables from $\bar{\mathrm{v}}$ — there are at most $2^{|\bar{\mathrm{v}}|^2}$ different choices for $\preceq$. We write $v_i \prec v_j$ if and only if we have $v_i \preceq v_j$ and $v_j \not\preceq v_i$. Let $v_{j_1} \prec \ldots \prec v_{j_k}$ be a maximal $\prec$-chain. Define the formula

$$\eta_\preceq(\bar{\mathrm{v}}) := \Big(\bigwedge_{v \prec v'} v < v' \Big) \wedge \Big(\bigwedge_{\substack{v \preceq v' \\ \wedge\, v' \preceq v}} v = v' \Big) \,. \qquad \eta_\preceq(\bar{\mathrm{v}})$$

(2) Find a tuple $\bar{r} \in \mathbb{Q}^{|\bar{\mathrm{v}}|}$ such that $\mathbb{Q} \models \varphi_1(\bar{r}) \wedge \eta_\preceq(\bar{r})$.

(3) Check whether the BSR sentence $\exists \bar{\mathrm{v}} \bar{\mathrm{y}} \forall \bar{\mathrm{z}}.\, \varphi_2(\bar{\mathrm{v}}, \bar{\mathrm{y}}, \bar{\mathrm{z}}) \wedge \eta_\preceq(\bar{\mathrm{v}})$ (with a mixture of interpreted and uninterpreted function and predicate symbols) is satisfiable, using the methods from Section 10.2, Lemma 10.2.11 in particular. If the answer is positive, construct a model $\mathcal{A} \models \exists \bar{\mathrm{v}} \bar{\mathrm{y}} \forall \bar{\mathrm{z}}.\, \varphi_2(\bar{\mathrm{v}}, \bar{\mathrm{y}}, \bar{\mathrm{z}}) \wedge \eta_\preceq(\bar{\mathrm{v}})$.

(4) If both Steps (2) and Step (3) succeed, then φ is satisfiable. More precisely, we then get $\mathcal{A} \models \varphi_1(\bar{r}) \wedge \big(\exists \bar{\mathrm{y}} \forall \bar{\mathrm{z}}.\, \varphi_2(\bar{r}, \bar{\mathrm{y}}, \bar{\mathrm{z}})\big) \wedge \eta_\preceq(\bar{r})$.

Notice that in this scheme the Steps (2) and (3) are independent of each other and only linked by the formula $\eta_\preceq$, which is based on the nondeterministically constructed preorder $\preceq$. From the perspective of φ_2, the variables in $\bar{\mathrm{v}}$ are constants whose exact values are unknown and not important. In fact, most parts of the structure $\mathbb{Q}$ are not important for constructing a model for φ_2. However, in general we have to make sure that any cardinality constraints that might be imposed by φ_2 are not in conflict with the fact that $<$ and $\leq$ under $\mathbb{Q}$ are *dense linear orders without endpoints*. For example, a BSR sentence of the form $\chi := \forall z_1 z_2.\, z_1 \neq z_2 \rightarrow \bigvee_{i=1}^n \big(P_i(z_1) \leftrightarrow \neg P_i(z_2)\big)$ limits

the domain to 2^n elements. In this case, using the rational numbers as (sub)domain is not possible. Such cardinality conflicts are resolved in Lemma 10.2.11 by (a) conjoining the formulas ψ and $\eta_{\preceq}$ (defined on pages 239 and 239) and (b) requiring $\mathcal{J}_{\mathcal{A}}$-uniformity of the model $\mathcal{A}$. In case of χ, this means that $\mathcal{A} \models \eta_{\preceq}$ entails the existence of two distinct domain elements a, b with $\mathsf{a} <^{\mathcal{A}} \mathsf{b}$ that are indistinguishable with respect to their belonging to the sets $P_i^{\mathcal{A}}$, i.e. $\mathcal{A} \models \bigwedge_i \big(P_i(\mathsf{a}) \leftrightarrow P_i(\mathsf{b})\big)$. Hence, we get $\mathcal{A} \not\models \chi$.

Remark 10.3.1. *The combination approach outlined so far allows for a neat black-box-style integration of arithmetic solving with theorem proving for purely uninterpreted first-order logic, in particular for BSR. From a practical point of view, the combined decision procedure suffers from the drawback that the (nondeterministic) search for a suitable preorder does not take the information into account that the decision procedures employed in Steps (2) and (3) gather while trying to solve the two parts $\exists \bar{\mathrm{v}}.\, \varphi_1(\bar{\mathrm{v}}) \wedge \eta_{\preceq}(\bar{\mathrm{v}})$ and $\exists \bar{\mathrm{v}} \bar{\mathrm{y}} \forall \bar{\mathrm{z}}.\, \varphi_2(\bar{\mathrm{v}}, \bar{\mathrm{y}}, \bar{\mathrm{z}}) \wedge \eta_{\preceq}(\bar{\mathrm{v}})$ individually. If we aim at combined decision procedures that are more efficient in practice, then a tighter cooperation of the component decision procedures is desirable, which could lead to a more directed search for the preorder $\preceq$.*

Consider again the sentence $\varphi = \exists \bar{\mathrm{v}}.\, \varphi_1(\bar{\mathrm{v}}) \wedge \exists \bar{\mathrm{y}} \forall \bar{\mathrm{z}}.\, \varphi_2(\bar{\mathrm{v}}, \bar{\mathrm{y}}, \bar{\mathrm{z}})$. In the light of the above said, it becomes clear that the constituent $\varphi_1(\bar{\mathrm{v}})$ does not necessarily have to be a linear-arithmetic sentence without quantifiers. In fact, the only requirement that is necessary for the described combination approach to work is the availability of a procedure that is able to provide us with a solution $\bar{r}$ for the variables $\bar{\mathrm{v}}$ such that $\varphi_1(\bar{r})$ is a valid arithmetic statement. Indeed, there are such procedures available for formulas $\varphi_1(\bar{\mathrm{v}})$ over the language of linear rational arithmetic with additional quantifiers and, more generally, for linear arithmetic over *ordered fields*. We have met one such a procedure based on quantifier elimination, namely *virtual substitution* [Wei88, LW93] in Section 7.1. When we consider the reals as domain, we could even allow polynomials instead of linear terms only — in this case, we are restricted to the model class of *real closed fields*.[5] For this language there are also quantifier-elimination procedures known that generalize the ones for the linear case, see [Stu17] for an overview. Regarding the second constituent $\exists \bar{\mathrm{y}} \forall \bar{\mathrm{z}}.\, \varphi_2(\bar{\mathrm{v}}, \bar{\mathrm{y}}, \bar{\mathrm{z}})$ of φ, we have focused on BSR sentences until now. But in the light of our insights gained in Chapter 3, in particular in Sections 3.2 and 3.5, it becomes clear that we could also use SF or GBSR formulas here.

Theorem 10.3.2. *Consider the class of first-order sentences of the form $\varphi = \exists \bar{\mathrm{v}}.\, \varphi_1(\bar{\mathrm{v}}) \wedge \varphi_2(\bar{\mathrm{v}})$ that satisfy the following properties.*

(a) $\varphi_1(\bar{\mathrm{v}})$ is a formula over the language of real arithmetic based on the vocabulary $\langle \{<, \leq, =, \neq, \geq, >\}, \mathbb{Q} \cup \{+, \cdot\} \rangle$. Then, all terms in $\varphi_1(\bar{\mathrm{v}})$ are polynomials over real-valued variables with rational coefficients.

(b) $\exists \bar{\mathrm{v}}.\, \varphi_2(\bar{\mathrm{v}})$ is a two-sorted GBSR sentence over the sort $\mathbb{R}$ and the uninterpreted sort $\mathcal{S}$. The underlying vocabulary contains the interpreted predicate symbols $<, \leq, =, \neq, \geq, >$ over the sort $\mathbb{R}$ and may also contain uninterpreted predicate symbols with signatures mixing the sorts $\mathbb{R}$ and $\mathcal{S}$. On the other hand, rational numbers or arithmetic operations such as $+, -, \cdot$ are not admitted in $\varphi_2(\bar{\mathrm{v}})$.

The satisfiability problem for the described class is decidable.

Proof sketch. The theorem follows from the fact that $\exists \bar{\mathrm{v}}.\, \varphi_2(\bar{\mathrm{v}})$ can be transformed into an equivalent BSR sentence (cf. Lemma 3.5.2) and the availability of first-order quantifier-elimination procedures for $\exists \bar{\mathrm{v}}.\, \varphi_1(\bar{\mathrm{v}}) \wedge \eta_{\preceq}(\bar{\mathrm{v}})$ for any preorder $\preceq$ on the variables in $\bar{\mathrm{v}}$. Together with these two components, the combined decision procedure described earlier suffices to solve the decision problem posed in the theorem. □

As explained above, the theorem is also valid if we replace $\mathbb{R}$ with $\mathbb{Q}$ under the restriction that all arithmetic terms in the constituent $\varphi_1(\bar{\mathrm{v}})$ are linear arithmetic terms.

[5] Recall that validity in the theory of the rationals with addition and multiplication is undecidable, cf. Footnote 2 on page 20.

By virtue of Theorem 10.3.2 and an adapted variant of Lemma 10.0.7, the following generalization of BSR(SLR) has a decidable satisfiability problem.

Definition 10.3.3 (GBSR with simple linear rational constraints — GBSR(SLR))**.** *A* GBSR(SLR) sentence *is any sentence of the form* $\varphi := \forall \bar{\mathrm{x}}_1 \exists \bar{\mathrm{y}}_1 \ldots \forall \bar{\mathrm{x}}_n \exists \bar{\mathrm{y}}_n . \psi$ *with quantifier-free* ψ *that adheres to Definition 3.4.1 and may contain LRA atoms that are subject to the following restriction. Every LRA atom in* φ *has the form* $s \lhd t$ *or* $x \lhd t$ *or* $x \lhd x'$ *where* $x, x' \in (\bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_n) \cap \mathrm{Var}_{\mathbb{Q}}$ *and* s, t *are LRA terms that do not contain any variable from* $\bar{\mathrm{x}}_1 \cup \ldots \cup \bar{\mathrm{x}}_n$ *but may contain variables from* $(\bar{\mathrm{y}}_1 \cup \ldots \cup \bar{\mathrm{y}}_n) \cap \mathrm{Var}_{\mathbb{Q}}$*. Every non-arithmetic atom in* φ *is either an equation* $s \approx s'$ *over variables from* $\mathrm{Var}_{\mathcal{S}}$*, or a non-equational atom* $P(s_1, \ldots, s_m)$ *that is well sorted and where the* s_i *range over base-sort variables and free-sort variables.*

Corollary 10.3.4. *The satisfiability problem for the class of GBSR(SLR) sentences is decidable.*

10.4 The Bernays–Schönfinkel–Ramsey Fragment with Bounded Difference Constraints is Decidable

Similarly to the previous section, we fix some finite BSR(BD) clause set N in BSR(BD) normal form, and we assume that all uninterpreted predicate symbols P occurring in N have the sort $P : \mathcal{S}^{m'} \times \mathbb{Q}^m$ for two fixed nonnegative integers m, m'. Moreover, we assume that all rational numbers in N are in fact integers. This does not lead to a loss of generality, as we could multiply all rational numbers with the least common multiple of their denominators to obtain an equisatisfiable clause set in which only integers occur. We could even allow Skolem constants, if we were to add clauses stipulating that every such constant is assigned a value that is (a) an integer and (b) is bounded from above and below by some integer bounds. Dropping any of these two restrictions renders the satisfiability problem undecidable, cf. Chapter 11, Sections 11.2 and 11.4 in particular. For the sake of simplicity, we do not consider Skolem constants in this section. N m, m'

Our general approach to proving decidability of the satisfiability problem for finite BSR(BD) clause sets is very similar to the route taken for BSR(SLR) in the previous section. However, due to the nature of the LRA atoms in BSR(BD) clause sets, the employed equivalence relation characterizing indistinguishable tuples has to be different from the one tailor-made for BSR(SLR). In fact, we use one equivalence relation $\widehat{\simeq}_\kappa$ over the unbounded space $\mathbb{Q}^m$ and another equivalence relation $\simeq_\kappa$ over the bounded subspace $(-\kappa - 1, \kappa + 1)^m$ for some positive integer κ. Our definition of the relations $\simeq_\kappa$ and $\widehat{\simeq}_\kappa$ is inspired by the notion of clock equivalence used in the context of timed automata (see, e.g., [AD94, BK08], and Section 10.5 of the present thesis).

Definition 10.4.1 (bounded region equivalence $\simeq_\kappa$)**.** *Let* κ *be any positive integer. We define the equivalence relation* $\simeq_\kappa$ *on* $(-\kappa - 1, \kappa + 1)^m$ *such that we get* $\langle r_1, \ldots, r_m \rangle \simeq_\kappa \langle s_1, \ldots, s_m \rangle$ *if and only if the following conditions are met:* $\simeq_\kappa$
(i) For every i *we have* $\lfloor r_i \rfloor = \lfloor s_i \rfloor$*, and* $fr(r_i) = 0$ *if and only if* $fr(s_i) = 0$*.*
(ii) For all i, j *we have* $fr(r_i) \leq fr(r_j)$ *if and only if* $fr(s_i) \leq fr(s_j)$*.*

The relation $\simeq_\kappa$ induces only a finite number of equivalence classes over $(-\kappa - 1, \kappa + 1)^m$. Over $\mathbb{Q}^m$, on the other hand, an analogous equivalence relation $\simeq_\infty$ would lead to infinitely many equivalence classes. In order to overcome this problem and obtain an equivalence relation over $\mathbb{Q}^m$ that induces only a finite number of equivalence classes, we use the following compromise.

Definition 10.4.2 (unbounded region equivalence $\widehat{\simeq}_\kappa$)**.** *Let* κ *be any positive integer. We define the equivalence relation* $\widehat{\simeq}_\kappa$ *on* $\mathbb{Q}^m$ *in such a way that we have* $\langle r_1, \ldots, r_m \rangle \widehat{\simeq}_\kappa \langle s_1, \ldots, s_m \rangle$ *if and only if* $\widehat{\simeq}_\kappa$

(i) for every i *either* $r_i > \kappa$ *and* $s_i > \kappa$*, or* $r_i < -\kappa$ *and* $s_i < -\kappa$*, or the following conditions are met:*

(i.i) $\lfloor r_i \rfloor = \lfloor s_i \rfloor$ *and*

(i.ii) $fr(r_i) = 0$ *if and only if* $fr(s_i) = 0$,

and

(ii) for all i, j

(ii.i) if $r_i, r_j > \kappa$ *or* $r_i, r_j < -\kappa$, *then* $r_i \leq r_j$ *if and only if* $s_i \leq s_j$,
(ii.ii) if $-\kappa \leq r_i, r_j \leq \kappa$, *then* $fr(r_i) \leq fr(r_j)$ *if and only if* $fr(s_i) \leq fr(s_j)$.

Obviously, the equivalence relations $\simeq_\kappa$ and $\widehat{\simeq}_\kappa$ coincide on the subspace $(-\kappa, \kappa)^m$. Over $(-\kappa - 1, \kappa + 1)^m$ the relation $\simeq_\kappa$ constitutes a proper refinement of $\widehat{\simeq}_\kappa$. Figure 10.2 depicts the equivalence classes induced by $\simeq_\kappa$ and $\widehat{\simeq}_\kappa$ in a two-dimensional setting for $\kappa = 1$. We will need both relations in our approach.

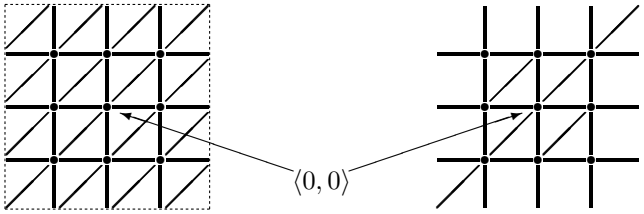


Figure 10.2: Left-hand side: partition of the set $(-2, 2)^2$ induced by $\simeq_1$. Right-hand side: partition of $\mathbb{Q}^2$ induced by $\widehat{\simeq}_1$. Every dot, line segment, and white area represents an equivalence class.

Definition 10.4.3 ($\simeq_\kappa$-uniform and $\widehat{\simeq}_\kappa$-uniform structures)**.** *Let* κ *be any positive integer. We call any structure* $\mathcal{A}$ $\simeq_\kappa$-uniform *if its corresponding coloring* $\chi_\mathcal{A}$ *(cf. Definition 10.2.5) colors each* $\simeq_\kappa$*-equivalence class over* $(-\kappa - 1, \kappa + 1)^m$ *uniformly, i.e. for all tuples* $\bar{q}, \bar{q}' \in (-\kappa - 1, \kappa + 1)^m$ *with* $\bar{q} \simeq_\kappa \bar{q}'$ *we require* $\chi_\mathcal{A}(\bar{q}) = \chi_\mathcal{A}(\bar{q}')$. *We call* $\mathcal{A}$ $\widehat{\simeq}_\kappa$-uniform *if* $\chi_\mathcal{A}$ *colors each* $\simeq_\kappa$*-equivalence class over* $\mathbb{Q}^m$ *uniformly.*

The parameter κ will be determined by the rational number in N with the largest absolute value. If κ is defined in this way, one can show that the LRA atoms occurring in N cannot distinguish between two $\widehat{\simeq}_\kappa$-equivalent m-tuples of rationals. This observation will be crucial for the proof of Lemma 10.4.4, where we will prove the existence of $\widehat{\simeq}_\kappa$-uniform models for satisfiable finite BSR(BD) clause sets N. To this end, we start from some model $\mathcal{A}$ of N and rely on the existence of a certain finite set $Q \subseteq [0, 1)$ of fractional parts. This set Q can be expanded to a set $\widehat{Q} \subseteq (-\kappa - 1, \kappa + 1)$ by addition of the fractional parts in Q with integral parts k from the range $-\kappa - 1 \leq k \leq \kappa$. Hence, $\widehat{Q}$ contains $2(\kappa+1) \cdot |Q|$ rational numbers. We assume that all $\simeq_\kappa$-equivalent tuples $\bar{s}, \bar{s}'$ from $\widehat{Q}^m$ are treated uniformly by $\mathcal{A}$. Put differently, we require $\chi_\mathcal{A}(\bar{s}) = \chi_\mathcal{A}(\bar{s}')$. We choose to formulate this requirement with respect to $\simeq_\kappa$ because of the more regular structure of its equivalence classes, which facilitates a more convenient way of invoking Lemma 10.1.1. Due to the fact that $\simeq_\kappa$ constitutes a refinement of $\widehat{\simeq}_\kappa$ on the subspace $(-\kappa - 1, \kappa + 1)^m$, and since for every $\widehat{\simeq}_\kappa$-equivalence class $\widehat{S}$ over $\mathbb{Q}^m$ there is some $\simeq_\kappa$-equivalence class $S \subseteq (-\kappa - 1, \kappa + 1)^m$ such that $S \subseteq \widehat{S}$, we can use the color $\chi_\mathcal{A}(\bar{r})$ of representative m-tuples $\bar{r}$ constructed from $\widehat{Q}$ to serve as a blueprint when constructing a $\widehat{\simeq}_\kappa$-uniform model $\mathcal{B}$.

λ, $\mathcal{A}$, κ **Lemma 10.4.4.** *Let* λ *be the maximal number of distinct base-sort variables in any single clause in* N*; in case of* $\lambda < m$, *we set* $\lambda := m$. *Suppose that* N *is satisfiable and let* $\mathcal{A}$ *be any model of* N. *Let* κ *be the smallest positive integer that is larger than the absolute value of any rational number*
Q *occurring in* N. *Suppose we are given a finite set* $Q \subseteq [0, 1)$ *of cardinality* $\lambda + 1$ *such that* $0 \in Q$ *and for all tuples* $\bar{r}, \bar{s} \in \widehat{Q}^m$, $\bar{r} \simeq_\kappa \bar{s}$ *entails* $\chi_\mathcal{A}(\bar{r}) = \chi_\mathcal{A}(\bar{s})$, *where*
$\widehat{Q}$
$$\widehat{Q} := \big\{ q + k \;\big|\; q \in Q \text{ and } k \text{ is any integer with } -\kappa - 1 \leq k \leq \kappa \big\}.$$
Then, we can construct a model $\mathcal{B}$ *of* N *that is* $\widehat{\simeq}_\kappa$*-uniform and that interprets the free sort* $\mathcal{S}$ *with some finite set.*

Proof. The construction of $\mathcal{B}$ from $\mathcal{A}$ is similar to the construction of uniform models outlined in the proof of Lemma 10.2.7.

Claim I: Let μ be any positive integer with $1 \leq \mu \leq \lambda$. For each of the finitely many equivalence classes $S \in \mathbb{Q}^\mu/_{\widehat{\simeq}_\kappa}$ and every $\bar{r} \in S$, there is some $\bar{q} \in S \cap \widehat{Q}^\mu$ such that $\bar{r} \mathrel{\widehat{\simeq}_\kappa} \bar{q}$ and for all i_1, i_2, i_3 with $r_{i_1} < -\kappa$ and $r_{i_2} > \kappa$ and $-\kappa \leq r_{i_3} \leq \kappa$ and $\mathrm{fr}(r_{i_3}) > 0$ we have $\mathrm{fr}(q_{i_1}) < \mathrm{fr}(q_{i_2}) < \mathrm{fr}(q_{i_3})$. $\bar{r}$

Proof: Let $i_1, i_2, \ldots$ be all the indices from $\{1, \ldots, \mu\}$ for which we have $r_{i_j} > \kappa$ for every j. Analogously, let $\ell_1, \ell_2, \ldots$ be all the indices from $\{1, \ldots, \mu\}$ such that $r_{\ell_j} < -\kappa$ holds for every j. We define the rational number i_j, ℓ_j

$$\delta := \min\big(\big\{\mathrm{fr}(r_i) \;\big|\; -\kappa \leq r_i \leq \kappa \text{ and } \mathrm{fr}(r_i) > 0 \text{ and } 1 \leq i \leq m\big\} \cup \big\{\tfrac{1}{2}\big\}\big). \qquad \delta$$

There must be some integer t for which we get $-\frac{1}{2}\delta < \frac{1}{t} r_{\ell_j} < 0 < \frac{1}{t} r_{i_{j'}} < \frac{1}{2}\delta$ for all j, j'. Let $\bar{r}'$ be the tuple that we obtain from $\bar{r}$ by replacing every r_{i_j} with $\frac{1}{t} r_{i_j} + \frac{1}{2}\delta + \kappa$ and every r_{ℓ_j} with $\frac{1}{t} r_{\ell_j} + \frac{1}{2}\delta - \kappa - 1$. By construction, we observe $\bar{r}' \in (-\kappa - 1, \kappa + 1)^\mu$ and $\bar{r} \mathrel{\widehat{\simeq}_\kappa} \bar{r}'$. Moreover, we have $0 < \mathrm{fr}(\bar{r}'_{\ell_j}) < \frac{1}{2}\delta < \mathrm{fr}(\bar{r}'_{i_{j'}}) < \delta \leq \mathrm{fr}(r_k)$ for all j, j' and every r_k with $-\kappa \leq r_k \leq \kappa$ and $\mathrm{fr}(r_k) > 0$.

Next, we define the following ascending sequences

$s'_0 < s'_1 < \ldots < s'_k$, where $s'_0 = 0$ and the values s'_j with $j \geq 1$ are the strictly positive fractional parts in ascending order that occur in $\mathrm{fr}(\bar{r}')$, and

$q'_0 < q'_1 < \ldots < q'_\lambda$, which comprises all rationals in Q in ascending order, including $q'_0 = 0$.

We now construct a tuple $\bar{q} \in S \cap \widehat{Q}^\mu$ by setting $q_\ell := \lfloor r'_\ell \rfloor + q'_j$ for j such that $\mathrm{fr}(r'_\ell) = s'_j$. $\bar{q}$

Clearly, $\bar{r}'$ and $\bar{q}$ are $\simeq_\kappa$-equivalent. Since $\simeq_\kappa$ is a refinement of $\widehat{\simeq}_\kappa$ over the subspace $(-\kappa - 1, \kappa + 1)^\mu$, this entails $\bar{r} \mathrel{\widehat{\simeq}_\kappa} \bar{q}$. ◊

Let $c_1, \ldots, c_\ell$ be an enumeration of all constant symbols in N that are of the sort $\mathcal{S}$. Let $\widehat{\mathcal{S}}$ denote the set $\{\mathsf{a} \in \mathcal{S}^{\mathcal{A}} \mid \mathsf{a} = c_i^{\mathcal{A}} \text{ for some } c_i\}$. We construct the structure $\mathcal{B}$ as follows. We set $\widehat{\mathcal{S}}$, $\mathcal{B}$ $\mathcal{S}^{\mathcal{B}} := \widehat{\mathcal{S}}$, and for every constant symbol c occurring in N we set $c^{\mathcal{B}} := c^{\mathcal{A}}$. Furthermore, for every uninterpreted predicate symbol P occurring in N and for all tuples $\bar{\mathsf{a}} \in \widehat{\mathcal{S}}^{m'}$ and $\bar{r} \in \mathbb{Q}^m$ we pick some tuple $\bar{q} \in \widehat{Q}^m$ in accordance with Claim I — i.e. $\bar{q}$ satisfies $\bar{r} \mathrel{\widehat{\simeq}_\kappa} \bar{q}$ — and define $P^{\mathcal{B}}$ in such a way that

$$\langle \bar{\mathsf{a}}, \bar{r} \rangle \in P^{\mathcal{B}} \quad \text{if and only if} \quad \langle \bar{\mathsf{a}}, \bar{q} \rangle \in P^{\mathcal{A}} \,.$$

Claim II: The structure $\mathcal{B}$ is $\widehat{\simeq}_\kappa$-uniform.

Proof: Let $\bar{r}^1, \bar{r}^2 \in \mathbb{Q}^m$ be two $\widehat{\simeq}_\kappa$-equivalent tuples. By Claim I, there exist two tuples $\bar{q}^1, \bar{q}^2 \in \widehat{Q}^m$ such that $\bar{q}^1 \mathrel{\widehat{\simeq}_\kappa} \bar{r}^1$ and $\bar{q}^2 \mathrel{\widehat{\simeq}_\kappa} \bar{r}^2$. By transitivity and symmetry of $\widehat{\simeq}_\kappa$, we have $\bar{q}^1 \mathrel{\widehat{\simeq}_\kappa} \bar{q}^2$. Even stronger, we can infer $\bar{q}^1 \simeq_\kappa \bar{q}^2$ as follows. Suppose, $\bar{q}^1 \not\simeq_\kappa \bar{q}^2$. We observe the following properties, which follow from $\bar{q}^1 \mathrel{\widehat{\simeq}_\kappa} \bar{q}^2$ and the fact that $\bar{q}^1, \bar{q}^2 \in (-\kappa - 1, \kappa + 1)^m$: $\bar{r}^1$, $\bar{r}^2$, $\bar{q}^1$, $\bar{q}^2$

(i) $\lfloor \bar{q}^1 \rfloor = \lfloor \bar{q}^2 \rfloor$ and $\lceil \bar{q}^1 \rceil = \lceil \bar{q}^2 \rceil$.

(ii) For all i, j for which $-\kappa \leq q_i^1, q_j^1 \leq \kappa$ we have $\mathrm{fr}(q_i^1) \leq \mathrm{fr}(q_j^1)$ if and only if $\mathrm{fr}(q_i^2) \leq \mathrm{fr}(q_j^2)$.

(iii) For all i, j for which $q_i^1, q_j^1 < -\kappa$ or $\kappa < q_i^1, q_j^1$ we have $q_i^1 \leq q_j^1$ if and only if $q_i^2 \leq q_j^2$. Because of $\bar{q}^1, \bar{q}^2 \in (-\kappa - 1, \kappa + 1)^m$, we even obtain $\mathrm{fr}(q_i^1) \leq \mathrm{fr}(q_j^1)$ if and only if $\mathrm{fr}(q_i^2) \leq \mathrm{fr}(q_j^2)$.

Hence, our assumption $\bar{q}^1 \not\simeq_\kappa \bar{q}^2$ entails that there are two indices i, j such that $\mathrm{fr}(q_i^1) \leq \mathrm{fr}(q_j^1)$ and $\mathrm{fr}(q_i^2) > \mathrm{fr}(q_j^2)$ (or $\mathrm{fr}(q_i^1) < \mathrm{fr}(q_j^1)$ and $\mathrm{fr}(q_i^2) \geq \mathrm{fr}(q_j^2)$), and one of the following cases applies:

(1) $\kappa < q_i^1, q_i^2$ and $-\kappa \leq q_j^1, q_j^2 \leq \kappa$, or

(2) $\kappa < q_i^1, q_i^2$ and $q_j^1, q_j^2 < -\kappa$, or

(3) $-\kappa \leq q_i^1, q_i^2 \leq \kappa$ and $\kappa < q_j^1, q_j^2$, or

(4) $-\kappa \leq q_i^1, q_i^2 \leq \kappa$ and $q_j^1, q_j^2 < -\kappa$, or

(5) $q_i^1, q_i^2 < -\kappa$ and $-\kappa \leq q_j^1, q_j^2 \leq \kappa$, or

(6) $q_i^1, q_i^2 < -\kappa$ and $\kappa < q_j^1, q_j^2$.

Ad (1). By Claim I, we have $\mathrm{fr}(q_i^1) < \mathrm{fr}(q_j^1)$ and $\mathrm{fr}(q_i^2) < \mathrm{fr}(q_j^2)$.

Ad (2). By Claim I, we have $\mathrm{fr}(q_i^1) > \mathrm{fr}(q_j^1)$ and $\mathrm{fr}(q_i^2) > \mathrm{fr}(q_j^2)$.

Ad (3). By Claim I, we have $\mathrm{fr}(q_i^1) > \mathrm{fr}(q_j^1)$ and $\mathrm{fr}(q_i^2) > \mathrm{fr}(q_j^2)$.

Ad (4). By Claim I, we have $\mathrm{fr}(q_i^1) > \mathrm{fr}(q_j^1)$ and $\mathrm{fr}(q_i^2) > \mathrm{fr}(q_j^2)$.

Ad (5). By Claim I, we have $\mathrm{fr}(q_i^1) < \mathrm{fr}(q_j^1)$ and $\mathrm{fr}(q_i^2) < \mathrm{fr}(q_j^2)$.

Ad (6). By Claim I, we have $\mathrm{fr}(q_i^1) < \mathrm{fr}(q_j^1)$ and $\mathrm{fr}(q_i^2) < \mathrm{fr}(q_j^2)$.

Since all cases yield a contradiction, we must have $\bar{q}^1 \simeq_\kappa \bar{q}^2$.

Because of $\bar{q}^1, \bar{q}^2 \in \widehat{Q}^m$ and due to our assumptions regarding Q and $\widehat{Q}^m$, we have $\chi_{\mathcal{A}}(\bar{q}^1) = \chi_{\mathcal{A}}(\bar{q}^2)$. Moreover, by construction of $\mathcal{B}$, we have $\chi_{\mathcal{B}}(\bar{r}^1) = \chi_{\mathcal{A}}(\bar{q}^1)$ and $\chi_{\mathcal{B}}(\bar{r}^2) = \chi_{\mathcal{A}}(\bar{q}^2)$. Consequently, $\chi_{\mathcal{B}}(\bar{r}^1) = \chi_{\mathcal{B}}(\bar{r}^2)$. ◊

$C = \Lambda \wedge \Gamma \to \Delta$ We next show $\mathcal{B} \models N$. Consider any clause $C = \Lambda \wedge \Gamma \to \Delta$ in N and let β be any variable assignment ranging over $\mathcal{S}^{\mathcal{B}} \cup \mathbb{Q}$. Starting from β, we derive a special variable assignment γ_C as follows. Let $x_1, \ldots, x_\ell$ be an enumeration of all base-sort variables in C. By Claim I, there exists some tuple $\bar{q} := \langle q_1, \ldots, q_\ell \rangle$ such that $\langle q_1, \ldots, q_\ell \rangle \mathrel{\widehat{\simeq}}_\kappa \langle \beta(x_1), \ldots, \beta(x_\ell) \rangle$ and $\bar{q} \in \widehat{Q}^\ell$. We define $\gamma_C(x_i) := q_i$ for every i. Hence, we have

β

γ_C

$$\langle \gamma_C(x_1), \ldots, \gamma_C(x_\ell) \rangle \mathrel{\widehat{\simeq}}_\kappa \langle \beta(x_1), \ldots, \beta(x_\ell) \rangle \,. \tag{10.2}$$

For all other base-sort variables $y \notin \{x_1, \ldots, x_\ell\}$ we could define $\gamma_C(y)$ arbitrarily. For every free-sort variable u we set $\gamma_C(u) := \beta(u)$.

As $\mathcal{A}$ is a model of N, we know $\mathcal{A}, \gamma_C \models C$. By case distinction on why $\mathcal{A}, \gamma_C \models C$ holds, we use this result to infer $\mathcal{B}, \beta \models C$.

Case $\mathcal{A}, \gamma_C \not\models x \lhd c$ for some atom $x \lhd c$ in Λ. Hence, $\beta_C(x) \not\lhd c$. Due to Equivalence (10.2), the assumption $|c| \leq \kappa$, and the definition of $\mathrel{\widehat{\simeq}}_\kappa$, we know that $\gamma_C(x) \lhd c$ holds if and only if $\beta(x) \lhd c$. Consequently, we get $\beta(x) \not\lhd c$ and thus $\mathcal{B}, \beta \not\models x \lhd c$.

Case $\mathcal{A}, \gamma_C \not\models x \lhd y$ for some atom $x \lhd y$ in Λ. By Equivalence (10.2) and the definition of $\mathrel{\widehat{\simeq}}_\kappa$, we know that $\gamma_C(x) \lhd \gamma_C(y)$ if and only if $\beta(x) \lhd \beta(y)$. Consequently, we get $\mathcal{B}, \beta \not\models x \lhd y$.

Case $\mathcal{A}, \gamma_C \not\models x - y \lhd c$ for some atom $x - y \lhd c$ in Λ. By definition of BSR(BD) clause sets, Λ must also contain atoms $c_x \leq x$, $x \leq d_x$, $c_y \leq y$, and $y \leq d_y$ for certain rational numbers c_x, d_x, c_y, d_y whose absolute value is at most κ. If one of these atoms is not satisfied by γ_C, then the first case applies.

If all of these atoms are satisfied by γ_C, then, by Equivalence (10.2), they are also satisfied by β. Moreover, Equivalence (10.2) and the definition of $\mathrel{\widehat{\simeq}}_\kappa$, entail $\lfloor \gamma_C(x) \rfloor = \lfloor \beta(x) \rfloor$, $\lfloor \gamma_C(y) \rfloor = \lfloor \beta(y) \rfloor$, $\lceil \gamma_C(x) \rceil = \lceil \beta(x) \rceil$, $\lceil \gamma_C(y) \rceil = \lceil \beta(y) \rceil$, $\mathrm{fr}(\gamma_C(x)) \leq \mathrm{fr}(\gamma_C(y))$ if and only if $\mathrm{fr}(\beta(x)) \leq \mathrm{fr}(\beta(y))$, and $\mathrm{fr}(\gamma_C(x)) \geq \mathrm{fr}(\gamma_C(y))$ if and only if $\mathrm{fr}(\beta(x)) \geq \mathrm{fr}(\beta(y))$. Hence, the following two observations hold:

$$\begin{aligned} \lfloor \gamma_C(x) - \gamma_C(y) \rfloor &= \lfloor \gamma_C(x) \rfloor - \lfloor \gamma_C(y) \rfloor + \lfloor \mathrm{fr}(\gamma_C(x)) - \mathrm{fr}(\gamma_C(y)) \rfloor \\ &= \lfloor \beta(x) \rfloor - \lfloor \beta(y) \rfloor + \lfloor \mathrm{fr}(\beta(x)) - \mathrm{fr}(\beta(y)) \rfloor \\ &= \lfloor \beta(x) - \beta(y) \rfloor \end{aligned}$$

and

$$\begin{aligned}\lceil \gamma_C(x) - \gamma_C(y) \rceil &= \lceil \gamma_C(x) \rceil - \lceil \gamma_C(y) \rceil + \lceil \mathrm{fr}(\gamma_C(x)) - \mathrm{fr}(\gamma_C(y)) \rceil \\ &= \lceil \beta(x) \rceil - \lceil \beta(y) \rceil + \lceil \mathrm{fr}(\beta(x)) - \mathrm{fr}(\beta(y)) \rceil \\ &= \lceil \beta(x) - \beta(y) \rceil \ .\end{aligned}$$

Consequently, since we assume c to be an integer, we have $\gamma_C(x) - \gamma_C(y) \lhd c$ if and only if $\beta(x) - \beta(y) \lhd c$. In other words, $\mathcal{A}, \beta \not\models x - y \lhd c$.

Case $\mathcal{A}, \gamma_C \not\models t \approx t'$ for some atom $t \approx t' \in \Gamma$. Hence, t and t' are either variables or constant symbols of the free sort, which means they do not contain subterms of the base sort. Since $\mathcal{B}$ and $\mathcal{A}$ behave identical on free-sort constant symbols and $\beta(u) = \gamma_C(u)$ for every variable $u \in V_{\mathcal{S}}$, we get $\mathcal{B}, \beta \not\models t \approx t'$.

Case $\mathcal{A}, \gamma_C \models t \approx t'$ for some $t \approx t' \in \Delta$. In analogy to the above case, we obtain $\mathcal{B}, \beta \models t \approx t'$.

Case $\mathcal{A}, \gamma_C \not\models P(t'_1, \ldots, t'_{m'}, t_1, \ldots, t_m)$ for some non-equational atom $P(t'_1, \ldots, t'_{m'}, t_1, \ldots, t_m)$ in Γ. This translates to

$$\big\langle \mathcal{A}(\gamma_C)(t'_1), \ldots, \mathcal{A}(\gamma_C)(t'_{m'}), \mathcal{A}(\gamma_C)(t_1), \ldots, \mathcal{A}(\gamma_C)(t_m) \big\rangle \notin P^{\mathcal{A}}.$$

By construction of γ_C, we have $\mathcal{A}(\gamma_C)(t_j) \in \widehat{Q}$ for every j. Due to our assumptions regarding $\widehat{Q}$ and by construction of $\mathcal{B}$, we therefore have

$$\big\langle \mathcal{A}(\gamma_C)(t'_1), \ldots, \mathcal{A}(\gamma_C)(t'_{m'}), \mathcal{A}(\gamma_C)(t_1), \ldots, \mathcal{A}(\gamma_C)(t_m) \big\rangle \notin P^{\mathcal{B}}.$$

We observe the following properties:

We have $\mathcal{A}(\gamma_C)(t'_j) = \mathcal{B}(\beta)(t'_j)$ for every t_j due to the definition of $\mathcal{B}$ and γ_C.

Since all the t_j are base-sort variables, we get $\mathcal{A}(\gamma_C)(t_j) = \mathcal{B}(\gamma_C)(t_j)$ for every t_j.

These two observations yield

$$\big\langle \mathcal{B}(\beta)(t'_1), \ldots, \mathcal{B}(\beta)(t'_{m'}), \mathcal{B}(\gamma_C)(t_1), \ldots, \mathcal{B}(\gamma_C)(t_m) \big\rangle \notin P^{\mathcal{B}}.$$

Because of this result, and due to $\widehat{\simeq}_\kappa$-uniformity of $\mathcal{B}$,

$$\big\langle \mathcal{B}(\gamma_C)(t_1), \ldots, \mathcal{B}(\gamma_C)(t_m) \big\rangle \mathrel{\widehat{\simeq}_\kappa} \big\langle \mathcal{B}(\beta)(t_1), \ldots, \mathcal{B}(\beta)(t_m) \big\rangle$$

entails

$$\big\langle \mathcal{B}(\beta)(t'_1), \ldots, \mathcal{B}(\beta)(t'_{m'}), \mathcal{B}(\beta)(t_1), \ldots, \mathcal{B}(\beta)(t_m) \big\rangle \notin P^{\mathcal{B}}.$$

Put differently, we have $\mathcal{B}, \beta \not\models P(t'_1, \ldots, t'_{m'}, t_1, \ldots, t_m)$.

Case $\mathcal{A}, \gamma_C \models P(t'_1, \ldots, t'_{m'}, t_1, \ldots, t_m)$ for some non-equational atom $P(t'_1, \ldots, t'_{m'}, t_1, \ldots, t_m)$ in Δ. In analogy to the previous case we infer $\mathcal{B}, \beta \models P(t'_1, \ldots, t'_{m'}, t_1, \ldots, t_m)$.

Altogether, we have shown $\mathcal{B} \models N$. □

We shall employ Lemma 10.1.1 to prove the existence of the set Q that is required for Lemma 10.4.4. This will finish the proof concerning the existence of $\widehat{\simeq}_\kappa$-uniform models for finite satisfiable BSR(BD) clause sets. But first, we need the following auxiliary result.

Lemma 10.4.5. *Let $S \in (-\kappa - 1, \kappa + 1)^m / \simeq_\kappa$ be an equivalence class with respect to $\simeq_\kappa$. There are two mappings $\rho : [m] \to \{0, 1, \ldots, m\}$ and $\sigma : [m] \to \{-\kappa - 1, \ldots, 0, \ldots, \kappa\}$ such that* S

(i) for every ascending tuple $\langle r_0, r_1, \ldots, r_m \rangle \in [0, 1)^{m+1}$ with $r_0 = 0$ we have $\big\langle r_{\rho(1)} + \sigma(1), \ldots, r_{\rho(m)} + \sigma(m) \big\rangle \in S$, and

(ii) for every tuple $\langle s_1, \ldots, s_m \rangle \in S$ there is an ascending tuple $\langle r_0, r_1, \ldots, r_m \rangle \in [0, 1)^{m+1}$ with $r_0 = 0$ such that $\big\langle s_1, \ldots, s_m \big\rangle = \big\langle r_{\rho(1)} + \sigma(1), \ldots, r_{\rho(m)} + \sigma(m) \big\rangle$.

Proof. Fix some tuple $\bar{q}$ taken from S. Given $\bar{q}$, we set $q'_0 := 0$ and further construct the sequence $q'_1, q'_2, \ldots$ in such a way that it lists all strictly positive fractional values occurring in $\mathrm{fr}(\bar{q})$ in $\bar{q}$, q'_i

σ, ρ ascending order. We construct σ by setting $\sigma(i) := \lfloor q_i \rfloor$ for every i, and ρ such that $\rho(i) = k$ holds if and only if $\mathrm{fr}(q_i) = q'_k$. Consequently, we have

$$\langle q_1, \ldots, q_m \rangle = \big\langle \mathrm{fr}(q_1) + \lfloor q_1 \rfloor, \ldots, \mathrm{fr}(q_m) + \lfloor q_m \rfloor \big\rangle = \big\langle q'_{\rho(1)} + \sigma(1), \ldots, q'_{\rho(m)} + \sigma(m) \big\rangle \,. \tag{10.3}$$

r_i Let $\langle r_0, r_1, \ldots, r_m \rangle \in [0,1)^{m+1}$ be any ascending tuple with $r_0 = 0$. For all i, j we observe the following properties:

(1) $\lfloor r_{\rho(i)} + \sigma(i) \rfloor = \sigma(i) = \lfloor q_i \rfloor$.

(2) $\mathrm{fr}(r_{\rho(i)} + \sigma(i)) = \mathrm{fr}(r_{\rho(i)}) = r_{\rho(i)}$.

(3) $\rho(i) = 0$ if and only if $\mathrm{fr}(q_i) = q'_0 = 0$, which entails that $\mathrm{fr}(r_{\rho(i)} + \sigma(i)) = 0$ holds if and only if we have $\mathrm{fr}(q_i) = 0$.

(4) $\mathrm{fr}(q_i) = q'_{\rho(i)}$.

(5) We have $\mathrm{fr}(r_{\rho(i)} + \sigma(i)) \leq \mathrm{fr}(r_{\rho(j)} + \sigma(j))$
if and only if $r_{\rho(i)} \leq r_{\rho(j)}$
if and only if $\rho(i) \leq \rho(j)$
if and only if $q'_{\rho(i)} \leq q'_{\rho(j)}$
if and only if $\mathrm{fr}(q_i) \leq \mathrm{fr}(q_j)$.

Taken together, these observations imply $\bar{q} \simeq_\kappa \langle r_{\rho(1)} + \sigma(1), \ldots, r_{\rho(m)} + \sigma(m) \rangle$. Hence, we have just proved (i).

In fact, we have also already proved (ii), by giving the construction of the sequence $q'_0, q'_1, q'_2, \ldots$ and by having derived Equation (10.3). If the sequence $q'_1, q'_2, \ldots$ is shorter than m elements, we can simply pad it in an ascending fashion with arbitrary values from the interval $(0, 1)$. □

We now have all necessary auxiliary results in place to prove Lemma 10.4.6, which stipulates the existence of the set Q required by Lemma 10.4.4.

$\mathcal{A}$, κ, λ **Lemma 10.4.6.** *Let $\mathcal{A}$ be any structure and let κ, λ be positive integers with $\lambda \geq m$. There exists a finite set $Q \subseteq [0,1)$ of cardinality $\lambda + 1$ such that $0 \in Q$ and for all tuples $\bar{s}, \bar{s}' \in \widehat{Q}^m$, $\bar{s} \simeq_\kappa \bar{s}'$ entails $\chi_{\mathcal{A}}(\bar{s}) = \chi_{\mathcal{A}}(\bar{s}')$, where*

$$\widehat{Q} := \big\{ q + k \;\big|\; q \in Q \text{ and } k \text{ is any integer with } -\kappa - 1 \leq k \leq \kappa \big\}.$$

S_j *Proof.* Let $S_1, \ldots, S_k$ be some enumeration of all equivalence classes in $(-\kappa - 1, \kappa + 1)^m / \simeq_\kappa$. By
$\langle \rho_j, \sigma_j \rangle$ Lemma 10.4.5, there is a (not necessarily unique) sequence $\langle \rho_1, \sigma_1 \rangle, \ldots, \langle \rho_k, \sigma_k \rangle$ of pairs of mappings such that each pair $\langle \rho_j, \sigma_j \rangle$ corresponds to the equivalence class S_j in the sense of Lemma 10.4.5.

$\widehat{\mathcal{S}}$ Let $c_1, \ldots, c_\ell$ be an enumeration of all constant symbols in N that are of the sort $\mathcal{S}$. Let $\widehat{\mathcal{S}}$ denote the set $\{\mathsf{a} \in \mathcal{S}^{\mathcal{A}} \mid \mathsf{a} = c_i^{\mathcal{A}} \text{ for some } c_i\}$ containing all domain elements assigned to free-sort constant

$\widehat{\chi}$ symbols by $\mathcal{A}$. We define a coloring $\widehat{\chi} : \mathbb{Q}^m \to \big(\mathcal{P}\{P\bar{\mathsf{a}} \mid \bar{\mathsf{a}} \in \widehat{\mathcal{S}}^{m'} \text{ and } P \text{ occurs in } N\}\big)^k$ by setting

$$\widehat{\chi}(\bar{r}) := \big\langle \chi_{\mathcal{A}}\big(\langle r_{\rho_1(1)} + \sigma_1(1), \ldots, r_{\rho_1(m)} + \sigma_1(m) \rangle\big), \ldots, \chi_{\mathcal{A}}\big(\langle r_{\rho_k(1)} + \sigma_k(1), \ldots, r_{\rho_k(m)} + \sigma_k(m) \rangle\big) \big\rangle$$

for every tuple $\bar{r} = \langle r_1, \ldots, r_m \rangle \in (0,1)^m$, where we define r_0 to be 0. By virtue of Lemma 10.1.1,
Q', Q there is a set $Q' \subseteq (0,1)$ of cardinality λ such that all ascending tuples $\langle r_1, \ldots, r_m \rangle \in {Q'}^m$ are assigned the same color by χ. We then set $Q := Q' \cup \{0\}$.

Consider any equivalence class S_j and the corresponding pair $\langle \rho_j, \sigma_j \rangle$ and let $\bar{s}, \bar{s}' \in \widehat{Q}^m$ be two $\simeq_\kappa$-equivalent tuples. Let $q_1, q_2, \ldots$ be an enumeration of all the strictly positive fractional parts in $\mathrm{fr}(\bar{s})$ in ascending order and let $q_0 := 0$. Hence, $q_0 < q_1 < q_2 < \ldots$. By definition of ρ_j, σ_j, there are two ascending tuples $\bar{q} := \langle 0, q_1, \ldots, q_m \rangle$ and $\bar{q}' := \langle 0, q'_1, \ldots, q'_m \rangle$ in $[0,1)^{m+1}$ such that

$$\bar{s} = \langle q_{\rho_j(1)} + \sigma_j(1), \ldots, q_{\rho_j(m)} + \sigma_j(m) \rangle$$

and

$$\bar{s}' = \langle q'_{\rho_j(1)} + \sigma_j(1), \ldots, q'_{\rho_j(m)} + \sigma_j(m) \rangle.$$

Because of $\bar{s}, \bar{s}' \in \widehat{Q}^m$, we know that $\langle q_1, \ldots, q_m \rangle \in Q'^m$ and $\langle q'_1, \ldots, q'_m \rangle \in Q'^m$. Then, $\widehat{\chi}(\langle q_1, \ldots, q_m \rangle) = \widehat{\chi}(\langle q'_1, \ldots, q'_m \rangle)$ entails

$$\begin{aligned}\chi_{\mathcal{A}}(\bar{s}) &= \chi_{\mathcal{A}}\big(\langle q_{\rho_j(1)} + \sigma_j(1), \ldots, q_{\rho_j(m)} + \sigma_j(m) \rangle\big) \\ &= \chi_{\mathcal{A}}\big(\langle q'_{\rho_j(1)} + \sigma_j(1), \ldots, q'_{\rho_j(m)} + \sigma_j(m) \rangle\big) = \chi_{\mathcal{A}}(\bar{s}') \ . \end{aligned}$$ □

Lemmas 10.4.4 and 10.4.6 together entail the existence of $\widehat{\simeq}_\kappa$-uniform models for finite satisfiable BSR(BD) clause sets, where κ is defined like in Lemma 10.4.4.

Corollary 10.4.7. *Let κ be the smallest positive integer that is larger than the absolute value of any rational number occurring in N. If N is satisfiable, then it has a model $\mathcal{A}$ that is $\widehat{\simeq}_\kappa$-uniform and whose interpretation of the sort $\mathcal{S}$ is some finite set.*

Similarly to the BSR(SLR) case, Lemma 10.4.4 and Corollary 10.4.7 do not immediately lend themselves to constructing a decision procedure for finite BSR(BD) clause sets. We need results that are easier to handle computationally. To this end, we reuse some ideas that we have already presented in the context of BSR(SLR). Let κ be the smallest positive integer that is larger than the absolute value of any rational number occurring in N. Let λ be the maximal number of distinct base-sort variables in any single clause in N; in case of $\lambda < m$ we set $\lambda := m$. Let Var_κ be a set of first-order variables defined by

κ, λ

Var_κ

$$\mathrm{Var}_\kappa := \big\{ z_{q_i+k} \;\big|\; \text{where } q_i + k \text{ is a formal term for any pair of integers } i, k \text{ with } 0 \leq i \leq \lambda \text{ and } -\kappa - 1 \leq k \leq \kappa \big\} \ .$$

Then, Var_κ contains $(\lambda + 1) \cdot (2\kappa + 2)$ variables, each of which is intended to represent one value from the set $\widehat{Q}$, defined in Lemma 10.4.4. Let $\bar{\mathrm{z}}$ be a tuple listing all variables from Var_κ in any order. Moreover, let $d_1, \ldots, d_\ell$ be an enumeration of all free-sort constant symbols occurring in N. We construct a formula $\eta_\kappa(\bar{\mathrm{z}})$ (also containing the constant symbols $d_1, \ldots, d_\ell$) that has the following properties:

$\bar{\mathrm{z}}$

d_j

$\eta_\kappa(\bar{\mathrm{z}})$

(a) $\eta_\kappa(\bar{\mathrm{z}})$ contains the axioms of (strict) linear orders for $<, \leq$ (treated as uninterpreted predicate symbols),

(b) $\eta_\kappa(\bar{\mathrm{z}})$ introduces two fresh unary uninterpreted predicate symbols Rat and Free to represent the sorts $\mathbb{Q}$ and $\mathcal{S}$, respectively, and makes sure that the constant symbols d_j are assigned to sort $\mathcal{S}$ and that the variables in $\bar{\mathrm{z}}$ are assigned to sort $\mathbb{Q}$, and

(c) $\eta_\kappa(\bar{\mathrm{z}})$ introduces fresh binary uninterpreted predicate symbols $P_{x-y \lhd k}$ with $-\kappa - 1 \leq k \leq \kappa$ and $\lhd \in \{<, \leq, =, \neq, \geq, >\}$ which are intended to represent the predicates $x - y \lhd k$ by uninterpreted predicate symbols, and $\eta_\kappa(\bar{\mathrm{z}})$ makes sure that the $P_{x-y \lhd k}$ are defined in the intended way over the elements represented by the variables in $\bar{\mathrm{z}}$.

Formally, we define η_κ as follows:

$$\begin{aligned}\eta_\kappa(\bar{\mathrm{z}}) := \ & \big(\forall v.\, \mathrm{Rat}(v) \leftrightarrow \neg \mathrm{Free}(v)\big) \wedge \big(\forall xy.\, x < y \vee x \leq y \rightarrow \mathrm{Rat}(x) \wedge \mathrm{Rat}(y)\big) \\ & \wedge \bigwedge_{\substack{-\kappa-1 \leq k \leq \kappa \\ \lhd \in \{<, \leq, =, \neq, \geq, >\}}} \big(\forall xy.\, P_{x-y \lhd k}(x, y) \rightarrow \mathrm{Rat}(x) \wedge \mathrm{Rat}(y)\big) \\ & \wedge \bigwedge_{z \in \bar{\mathrm{z}}} \mathrm{Rat}(z) \wedge \bigwedge_{1 \leq j \leq \ell} \mathrm{Free}(d_j) \\ & \wedge \Big(\forall xyw.\, \mathrm{Rat}(x) \wedge \mathrm{Rat}(y) \wedge \mathrm{Rat}(w) \longrightarrow \Big(\big(\neg x < x\big) \\ & \qquad \wedge \big(x < y \wedge y < w \rightarrow x < w\big) \\ & \qquad \wedge \big(x \approx y \vee x < y \vee x < y\big) \\ & \qquad \wedge \big(x \leq y \leftrightarrow x \approx y \vee x < y\big)\Big)\Big) \\ & \wedge \bigwedge_{-\kappa-1 \leq k \leq \kappa} \Big(\Big(\bigwedge_{0 \leq j \leq \lambda-1} z_{q_j+k} < z_{q_{j+1}+k}\Big) \wedge z_{q_\lambda+k} < z_{q_0+k+1}\Big) \ . \end{aligned}$$

$\eta'_\kappa(\bar{z})$

The length of η_κ is polynomial in the number of variables in Var_κ, in κ and λ, and in the number of free-sort constant symbols occurring in N. In addition to η_κ, we define $\eta'_\kappa(\bar{z})$ to be a first-order formula stipulating the order axioms for the fresh predicate symbols $P_{x-y\lhd k}$. That is, $\eta_\kappa(\bar{z})$ in conjunction with $\eta'_\kappa(\bar{z})$ satisfies the following properties for all i, j, j', k, k' with $-\kappa \leq i \leq \kappa$ and $0 \leq j, j' \leq \lambda$ and $-\kappa - 1 \leq k, k' \leq \kappa$:

$\eta_\kappa(\bar{z}) \wedge \eta'_\kappa(\bar{z}) \models P_{x-y=i}(z_{q_j+k}, z_{q_{j'}+k'})$ if and only if $k - k' = i$ and $j = j'$,

$\eta_\kappa(\bar{z}) \wedge \eta'_\kappa(\bar{z}) \models P_{x-y<i}(z_{q_j+k}, z_{q_{j'}+k'})$ if and only if $k - k' < i$ or $k - k' = i$ and $j < j'$,

$\eta_\kappa(\bar{z}) \wedge \eta'_\kappa(\bar{z}) \models P_{x-y>i}(z_{q_j+k}, z_{q_{j'}+k'})$ if and only if $k - k' > i$ or $k - k' = i$ and $j > j'$,

$\eta_\kappa(\bar{z}) \wedge \eta'_\kappa(\bar{z}) \models P_{x-y\leq i}(z_{q_j+k}, z_{q_{j'}+k'})$ if and only if $\eta_\kappa(\bar{z}) \wedge \eta'_\kappa(\bar{z}) \models P_{x-y<i}(z_{q_j+k}, z_{q_{j'}+k'})$ or $\eta_\kappa(\bar{z}) \wedge \eta'_\kappa(\bar{z}) \models P_{x-y=i}(z_{q_j+k}, z_{q_{j'}+k'})$,

$\eta_\kappa(\bar{z}) \wedge \eta'_\kappa(\bar{z}) \models P_{x-y\geq i}(z_{q_j+k}, z_{q_{j'}+k'})$ if and only if $\eta_\kappa(\bar{z}) \wedge \eta'_\kappa(\bar{z}) \models P_{x-y>i}(z_{q_j+k}, z_{q_{j'}+k'})$ or $\eta_\kappa(\bar{z}) \wedge \eta'_\kappa(\bar{z}) \models P_{x-y=i}(z_{q_j+k}, z_{q_{j'}+k'})$,

$\eta_\kappa(\bar{z}) \wedge \eta'_\kappa(\bar{z}) \models P_{x-y\neq i}(z_{q_j+k}, z_{q_{j'}+k'})$ if and only if $\eta_\kappa(\bar{z}) \wedge \eta'_\kappa(\bar{z}) \models P_{x-y<i}(z_{q_j+k}, z_{q_{j'}+k'})$ or $\eta_\kappa(\bar{z}) \wedge \eta'_\kappa(\bar{z}) \models P_{x-y>i}(z_{q_j+k}, z_{q_{j'}+k'})$,

where all predicate symbols in $\eta_\kappa(\bar{z}) \wedge \eta'_\kappa(\bar{z})$ are treated as uninterpreted in the current context. The length of η'_κ is at most polynomial in $|\mathrm{Var}_\kappa|$ and κ — we could simply specify for every quadruple $z_{q_j+k}, z_{q_{j'}+k'}, \lhd, i$ whether $P_{x-y\lhd i}(z_{q_j+k}, z_{q_{j'}+k'})$ is satisfied or not. We assume, without loss of generality, that none of the predicate symbols Rat, Free, and $P_{x-y\lhd k}$ occur in N. Moreover, we assume that none of the variables from Var_κ occur in N. Let $\bar{u}$ be some tuple listing all variables from $\mathrm{vars}(N) \cap \mathrm{Var}_\mathcal{S}$ and let $\bar{x}$ be some tuple listing all variables from $\mathrm{vars}(N) \cap \mathrm{Var}_\mathbb{Q}$. Recall that we assume that all rational numbers in the BSR(BD) clause set N are integers. Let N' be the result of replacing every integer k in N with the variable z_{q_0+k}. We now define the sentence

N'

$\varphi_{N,\kappa}$

$$\varphi_{N,\kappa} := \exists \bar{z}.\, \eta_\kappa(\bar{z}) \wedge \eta'_\kappa(\bar{z}) \wedge \forall \bar{u}\bar{x}.\, \Big(\bigwedge_{u \in \bar{u}} \mathrm{Free}(u) \wedge \bigwedge_{x \in \bar{x}} \mathrm{Rat}(x) \Big) \rightarrow \bigwedge_{C(\bar{z},\bar{u},\bar{x}) \in N'} C(\bar{z}, \bar{u}, \bar{x}) \ ,$$

which is evidently equivalent to some BSR sentence. Furthermore, we observe that the length of $\varphi_{N,\kappa}$ is polynomial in κ, λ, and the length of N. For the following variant of Lemma 10.4.4 we again make an exception for the sentences $\varphi_{N,\kappa}$ and treat $<, \leq$ as uninterpreted predicate symbols and consider the sentence without sorts.

Lemma 10.4.8. *Suppose there is a model $\mathcal{A} \models \varphi_{N,\kappa}$ with a single-sorted domain and in which $<, \leq$ and the $P_{x-y\lhd k}$ are treated as uninterpreted predicate symbols. Assume that $\mathcal{A}$'s domain is minimal, i.e. $\mathcal{A}$ does not contain any substructure that also satisfies $\varphi_{N,\kappa}$ — notice that this entails that A is finite. Furthermore, assume that for all m-tuples $\bar{q}, \bar{q}'$ of elements from $\mathrm{Rat}^\mathcal{A}$ that are $\widehat{\simeq}_\kappa$-equivalent[6] we have $\chi_\mathcal{A}(\bar{q}) = \chi_\mathcal{A}(\bar{q}')$.*

Then, we can construct a model $\mathcal{B}$ of N that is $\widehat{\simeq}_\kappa$-uniform, constains the rational numbers as subdomain, interprets the predicate symbols $<, \leq$ as the usual relations over the rationals, and interprets the free sort $\mathcal{S}$ with some finite set.

The proof of Lemma 10.4.8 proceeds along the same lines as the proof of Lemma 10.4.4 does. We only need to switch from the setting of N mixing interpreted arithmetic relations and difference constraints over the rationals with uninterpreted predicate symbols to the point of view of the purely uninterpreted setting of $\varphi_{N,\kappa}$. Moreover, Corollary 10.4.7 guarantees the existence of a model $\mathcal{A}$ as described in Lemma 10.4.8 whenever N is satisfiable (in the arithmetic setting).

[6] Although $\simeq_\kappa$-equivalence and $\widehat{\simeq}_\kappa$-equivalence and the coloring function $\chi_\mathcal{A}$ are technically defined for a different setting, we reuse the definitions in Lemma 10.4.8 and in Proposition 10.4.9 with their intended meaning without formally adapting them to the new setting.

Proposition 10.4.9. *If N has a model $\mathcal{B}$, then there is a $\widehat{\simeq}_\kappa$-uniform[7] model $\mathcal{A} \models \varphi_{N,\kappa}$ with a finite domain.*

Finally, we have all pieces together to devise a nondeterministic decision procedure for finite BSR(BD) clause sets N that proceeds as follows:

(I) Construct the sentence $\varphi_{N,\kappa}$ corresponding to N and transform it into prenex normal form with some $\exists^*\forall^*$ quantifier prefix. Suppose the result is of the form $\exists\bar{\mathrm{v}}\forall\bar{\mathrm{w}}.\,\psi(\bar{\mathrm{v}},\bar{\mathrm{w}})$ with quantifier-free $\psi(\bar{\mathrm{v}},\bar{\mathrm{w}})$. In what follows we treat all predicate symbols in this sentence as if they were uninterpreted.

(II) Nondeterministically construct a candidate model $\mathcal{A}$ such that

  (a) $\mathcal{A}$'s domain is minimal: we do not introduce more domain elements than necessary to assign suitable values to $\bar{\mathrm{v}}$ and the constant symbols occurring in $\varphi_{N,\kappa}$, and

  (b) $\mathcal{A}$ is $\widehat{\simeq}_\kappa$-uniform[8] with respect to the elements in $\mathrm{Rat}^{\mathcal{A}}$.

(III) Check whether $\mathcal{A}$ is indeed a model of $\varphi_{N,\kappa}$.

By Lemma 10.4.8 and Proposition 10.4.9, the procedure is a correct and complete decision procedure for the satisfiability problem for finite BSR(BD) clause sets. Concerning computational complexity, we observe the following: Step (I) can certainly be done in polynomial time with respect to κ and the length of N. As we may assume that the integers in N are encoded in binary, the dependence on κ leads to a runtime bound that is polynomial in $2^{\|N\|}$. Proposition 3.1.6 together with a modified variant of Proposition 5.0.1 — compare the discussion right before Theorem 10.2.14 on page 241 — entails that Steps (II) and (III) together can be done nondeterministically in time $p\big(n^k \cdot \mathrm{len}(\varphi_{N,\kappa})\big)$ where p is some polynomial in a single argument, $n := |\bar{\mathrm{z}}| + |\mathrm{consts}(\varphi_{N,\kappa})|$, and k denotes the number of universal quantifiers in $\varphi_{N,\kappa}$, which we may assume to be linear in the maximal number of variables in any clause in N. Since n is linear in $2^{\|N\|}$, k is linear in $\mathrm{len}(N)$, and $\mathrm{len}(\varphi_{N,\kappa})$ is polynomial in κ, λ and $\mathrm{len}(N)$ and, hence, polynomial in $2^{\|N\|}$, we in the end get that the satisfiability problem for finite BSR(BD) clause sets lies in NExpTime. It is even NExpTime-complete, since the subproblem BSR-Sat is already NExpTime-hard. n, k

Theorem 10.4.10. *Satisfiability of finite BSR(BD) clause sets is decidable, and for clause sets in BSR(BD) normal form the problem is* NExpTime*-complete.*

By virtue of Lemma 10.0.7, transforming any finite BSR(BD) clause set N into an equisatisfiable finite clause set N' in BSR(BD) normal form leads to a blowup that is such that (a) the length of N' is at most exponential in the length of N, (b) for any clause C in N' the number of variables occurring in C is not larger than the number of variables occurring in any clause in N, (c) every free-sort Skolem constant occurring in N' also occurs in N, and (d) the absolute value of any integer in N' is linear in $\kappa_N \cdot \lambda_N$, where κ_N is the smallest positive integer that is larger than the absolute value of any integer occurring in N and λ_N is the maximal number of variables occurring in any clause in N. Let n' be the number of existentially quantified variables plus the number of constant symbols occurring in $\varphi_{N',\kappa'}$. Then, we observe that n' is polynomial in $\kappa_N \cdot \lambda_N$ and $\mathrm{len}(N)$ and, hence, n' is polynomial in $2^{\|N\|}$. Let k' be the number of universal quantifiers occurring in $\varphi_{N',\kappa'}$. Then, k' is linear in λ_N, which in turn is smaller than $\mathrm{len}(N)$. Consequently, satisfiability of N can be checked nondeterministically in time that is bounded from above by $p\big(2^{(\mathrm{len}(N))^d+1}\big)$ for some polynomial p and some positive integer constant d.

Corollary 10.4.11. *The satisfiability problem for finite BSR(BD) clause sets is* NExpTime*-complete.*

[7] See Footnote 6 on page 252.
[8] See Footnote 6 on page 252.

10.5 An Application: Formalizing Reachability for Timed Automata in BSR(BD)

Timed automata (cf. Definition 10.5.3), introduced in the 1990s [AD90, AD94, Lew90, HNSY94], and extensions thereof are a well-established and widely-used formalism for modeling behavior of state-based real-time systems. See [BK08] for a gentle textbook exposition and see the very recent handbook articles [BFL$^+$18, DFPP18] for comprehensive surveys. In the present section, we consider the reachability problem for timed automata, which poses the question whether one can reach a certain set of states from the initial state, possibly under timing restrictions. The involved concepts will be defined below (Definitions 10.5.2–10.5.4). Our goal is to show that this problem can be formalized using finite BSR(BD) clause sets. The encoding we shall use will be a variant of an encoding devised by Fietzke and Weidenbach [FW12]. The central idea underlying the modification is that time progress does not have to be modeled as precisely as done in the original encoding. It is well known that the *clock constraints* in any given timed automaton induce finitely many *regions* in the space of clock valuations. We shall refer to these regions as *TA regions* (cf. Definition 10.5.5). If two clock valuations belong to one and the same TA region, they are indistinguishable by the automaton and its clock constraints. This leads to a more abstract point of view where only the reachability of TA regions matters. As we adopt this point of view, passage of time can be modeled as a movement from one TA region into reachable regions rather than the movement of a single point in the space of clock valuations to a ray of reachable points (cf. Figure 10.4 on page 257). It turns out that difference constraints are sufficient to formalize time progress in terms of TA regions. This approach will be made precise in Lemmas 10.5.9 and 10.5.10. Furthermore, it not hard to see that it is sufficient to consider a bounded subspace of the space $\mathbb{Q}^m_{\geq 0}$, if we intend to decide reachability for a timed automaton with m clocks (cf. Proposition 10.5.6). For every such automaton there exists a computable integer κ, depending on m and the integers occurring in clock constraints, such that any valuation $\bar{r}$ of the clocks can be projected to some valuation $\bar{r}'$ that is indistinguishable from $\bar{r}$ by the occurring clock constraints and that lies inside of the space $[0, \kappa + 1)^m$. This is the reason why bounded difference constraints suffice to formalize reachability.

Remark 10.5.1. *In [NMA$^+$02] an encoding of the reachability problem for timed automata in* difference logic *(Boolean combinations of difference constraints* without *uninterpreted predicate symbols) is given, which facilitates deciding bounded reachability for timed automata, i.e. the problem of reaching a given set of states within a bounded number of transition steps. When using BSR(BD) as a modeling language, we do not have to fix an upper bound on the number of steps a priori.*

There are also other encodings of the reachability problem for timed automata and related formalisms into linear arithmetic known, for instance [QSW17] (based on mixtures of Presburger arithmetic and linear rational arithmetic) and [CJ98, CJ99] (based on the additive theory of rationals or integers).

m, $\bar{\mathrm{x}}$ *clock variables*

We shall use the standard definitions for timed automata and related notions (see, e.g. [AD94, BK08, BFL$^+$18]). In what follows, we fix a positive integer m and a finite tuple $\bar{\mathrm{x}}$ of length m containing pairwise-distinct first-order variables that have sort $\mathbb{Q}$, called *clock variables* or *clocks* for short.

Definition 10.5.2 (Clock constraints)**.** *An* atomic clock constraint over $\bar{\mathrm{x}}$ *is an atom of the form* $\mathtt{true}$*,* $x \triangleleft c$*, or* $x - y \triangleleft c$*, where* $x, y \in \bar{\mathrm{x}}$*,* $\triangleleft \in \{<, \leq, =, \geq, >\}$*, and* $c \geq 0$ *is a nonnegative integer.*

$\mathrm{ACC}(\bar{\mathrm{x}})$ *By* $\mathrm{ACC}(\bar{\mathrm{x}})$ *we denote the* set of all atomic clock constraints over $\bar{\mathrm{x}}$.

A clock constraint over $\bar{\mathrm{x}}$ *is a finite conjunction* $\varphi = \varphi_1 \wedge \ldots \wedge \varphi_k$ *of atomic clock constraints* $\varphi_1, \ldots, \varphi_k \in \mathrm{ACC}(\bar{\mathrm{x}})$ *for some* $k \geq 1$*. We denote the* set of all clock constraints over $\bar{\mathrm{x}}$ *by* $\mathrm{CC}(\bar{\mathrm{x}})$.

$\mathrm{CC}(\bar{\mathrm{x}})$

Definition 10.5.3 (Timed automaton)**.** *A* timed automaton *is a tuple*

$$\mathfrak{A} := \left\langle \mathrm{Loc}, \ell_0, \bar{\mathrm{x}}, \left(\mathrm{inv}_\ell(\bar{\mathrm{x}})\right)_{\ell \in \mathrm{Loc}}, \mathcal{T} \right\rangle$$

where Loc *is a finite set of* locations *(i.e. control states);* $\ell_0 \in \mathrm{Loc}$ *is the* initial location*;* $\bar{\mathrm{x}}$ *is a*

tuple of clock variables; $\mathrm{inv}_\ell(\bar{\mathrm{x}}) \in \mathrm{CC}(\bar{\mathrm{x}})$ *is a clock constraint describing the* location invariant *of location* ℓ*;* $\mathcal{T} \subseteq \mathrm{Loc} \times \mathrm{CC}(\bar{\mathrm{x}}) \times \mathcal{P}(\bar{\mathrm{x}}) \times \mathrm{Loc}$ *is the* location transition relation *of the automaton, including* transition guards *with respect to clocks and the set of clocks that are being* reset to zero *whenever the transition is taken. In addition, we assume* $\mathbb{Q} \models \mathrm{inv}_{\ell_0}(\bar{0})$ *and, moreover, we assume that every clock constraint* $\psi(\bar{\mathrm{x}})$ *occurring in a timed automaton over* $\bar{\mathrm{x}}$ *is satisfiable under* $\mathbb{Q}$*, i.e. we have* $\mathbb{Q} \models \exists \bar{\mathrm{x}}.\, \psi(\bar{\mathrm{x}})$*.*

The latter property can be checked in polynomial time, as already mentioned right after Proposition 10.0.6. Since we will be concerned with the reachability problem only, we do not consider an alphabet of actions that could provide additional labels for transitions. Hence, we implicitly assume a one-letter alphabet, but the obtained results could easily be transferred to richer alphabets.

Notice that we allow atoms $x - y \triangleleft c$ in clock constraints. Such constraints are often referred to as *diagonal constraints* in the timed-automata literature. It is known that they do not add expressiveness to the formalism, as any timed automaton with diagonal constraints can be transformed into an equivalent timed automaton that does not contain any diagonal constraints (see [BPDG98], Section 4.2). Two timed automata are considered to be *equivalent*, if they accept the same (timed) language ([BPDG98], Section 2.2).

Although the control flow of any timed automaton can be described by finite means, the fact that clocks can assume infinitely many values yields an infinite state space. Formally, the semantics of a timed automaton is given by an infinite transition system.

Definition 10.5.4 (Semantic transition system of a timed automaton)**.** *The semantics of a timed automaton* $\mathfrak{A} := \langle \mathrm{Loc}, \ell_0, \bar{\mathrm{x}}, (\mathrm{inv}_\ell(\bar{\mathrm{x}}))_{\ell \in \mathrm{Loc}}, \mathcal{T} \rangle$ *is given by an infinite* state transition system $\mathrm{TS}(\mathfrak{A}) := \langle S, s_0, \hookrightarrow \rangle$ *with the following components:* $\mathrm{TS}(\mathfrak{A})$

$S := \mathrm{Loc} \times \mathbb{Q}^m_{\geq 0} = \{ \langle \ell, \bar{r} \rangle \mid \ell \in \mathrm{Loc}$ *and* $\bar{r} \in \mathbb{Q}^m_{\geq 0} \}$ *is the state space consisting of locations paired with* clock valuations *— such a valuation is a total mapping* $\bar{\mathrm{x}} \to \mathbb{Q}^m_{\geq 0}$ *assigning nonnegative reals to* m *clock variables;* *clock valuations*

$s_0 := \langle \ell_0, \bar{0} \rangle$ *is the* initial state*, where* $\bar{0}$ *denotes the tuple of length* m *containing all zeros;*

$\hookrightarrow \subseteq S \times S$ *is the* transition relation *containing two kinds of transitions:*

delay transitions $\{ \langle \ell, \bar{r} \rangle \hookrightarrow \langle \ell, \bar{r}' \rangle \mid \ell \in \mathrm{Loc}$ *and there is some* $t \geq 0$ *such that* $\bar{r}' := \bar{r} + t$ *and* $\mathbb{Q} \models \mathrm{inv}_\ell(\bar{r}') \}$*;*

location transitions $\{ \langle \ell, \bar{r} \rangle \hookrightarrow \langle \ell', \bar{r}' \rangle \mid$ *there is some* $\langle \ell, \psi(\bar{\mathrm{x}}), Z, \ell' \rangle \in \mathcal{T}$ *such that* $\mathbb{Q} \models \psi(\bar{r})$*,* $\bar{r}' := \bar{r}[Z \mapsto 0]$*, and* $\mathbb{Q} \models \mathrm{inv}_{\ell'}(\bar{r}')$*,*

where $\bar{r} + t$ *is the tuple* $\langle r_1 + t, \ldots, r_m + t \rangle$ *and* $\bar{r}[Z \mapsto 0]$ *stands for the tuple* $\bar{r}'$ *with*

$$r'_i := \begin{cases} 0 & \text{if } x_i \in Z, \\ \bar{r}_i & \text{if } x_i \notin Z \end{cases}$$

for every index i*.*

We denote the reflexive transitive closure of $\hookrightarrow$ *by* $\hookrightarrow^*$*.* $\hookrightarrow^*$

Any pair $\langle \ell, \bar{r} \rangle \in \mathrm{Loc} \times \mathbb{Q}^m$ *is called* reachable in $\mathfrak{A}$*, if we have* $s_0 \hookrightarrow^* \langle \ell, \bar{r} \rangle$*.* *reachable in* $\mathfrak{A}$

Consider any timed automaton $\mathfrak{A} := \langle \mathrm{Loc}, \ell_0, \bar{\mathrm{x}}, (\mathrm{inv}_\ell(\bar{\mathrm{x}}))_{\ell \in \mathrm{Loc}}, \mathcal{T} \rangle$. It is easy to verify that every clock constraint occurring in $\mathfrak{A}$ — be it as the initial condition, as transition invariant, or as transition guard — can be transformed into an equivalent conjunction ψ of difference constraints in the sense of Definition 10.0.5 and Proposition 10.0.6 (see the paragraph preceding Definition 10.0.5). Let κ be the smallest positive integer that is larger than the absolute value of any integer occurring in any clock constraint in $\mathfrak{A}$. Let $\mu := \kappa \cdot m$.[9] Since we assume all clock constraints in timed automata to be satisfiable under $\mathbb{Q}$, Proposition 10.0.6 entails that none of the clock constraints in $\mathfrak{A}$ can distinguish two clock valuations $\bar{r}, \bar{r}' \in [0, \mu + 1)^m$ that are $\simeq_\mu$-equivalent. On the one κ μ

[9] Notice that it is sufficient to set $\mu = \kappa \cdot m$ instead of $\mu := \kappa \cdot (m+1)$, as the start and end points of the paths we need to consider in difference constraint graphs associated with clock constraints in timed automata do not coincide.

hand, the following equivalence relation $\sim_{\mathfrak{A}}$ over $\mathbb{Q}_{\geq 0}$ is a refinement of $\widehat{\simeq}_\mu$ over $\mathbb{Q}_{\geq 0}$ and, on the other hand, $\simeq_\mu$ constitutes a refinement of $\sim_{\mathfrak{A}}$ over $[0, \mu+1)^m$.

Definition 10.5.5 ($\sim_{\mathfrak{A}}$, TA regions)**.** *Let $\mathfrak{A} := \langle \mathrm{Loc}, \ell_0, \bar{\mathrm{x}}, (\mathrm{inv}_\ell(\bar{\mathrm{x}}))_{\ell \in \mathrm{Loc}}, \mathcal{T} \rangle$ be a timed automaton. Let κ be the smallest positive integer that is larger than the absolute value of any integer occurring in any clock constraint in $\mathfrak{A}$ and let $\mu := \kappa \cdot m$. We define the equivalence relation $\sim_{\mathfrak{A}}$ on $\mathbb{Q}_{\geq 0}^m$ such that $\bar{r} \sim_{\mathfrak{A}} \bar{s}$ holds if and only if*

(i) for every i we either have $\lfloor r_i \rfloor = \lfloor s_i \rfloor$, or $r_i > \mu$ and $s_i > \mu$, and

(ii) for all i, j we either have $\lfloor r_i - r_j \rfloor = \lfloor s_i - s_j \rfloor$, or $r_i - r_j > \mu$ and $s_i - s_j > \mu$, or $r_i - r_j < -\mu$ and $s_i - s_j < -\mu$.

We call the equivalence classes induced by $\sim_{\mathfrak{A}}$ over $\mathbb{Q}_{\geq 0}^m$ the TA regions of $\mathfrak{A}$.

Figure 10.3 illustrates the TA regions for a timed automaton with two clocks and in which all integer constants have an absolute value of at most 2. For every TA region $R \subseteq \mathbb{Q}_{\geq 0}^2$ of such an automaton, there is at least one representative $\bar{r} \in R$ which lies in the subspace $[0, 5)^2$.

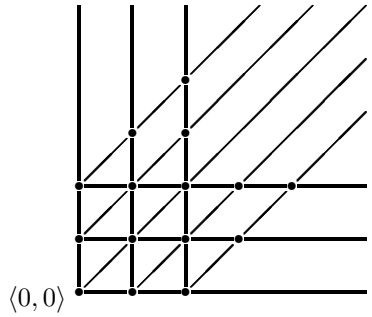


Figure 10.3: Partition of the set $\mathbb{Q}_{\geq 0}^2$ into $\sim_{\mathfrak{A}}$-equivalence classes of clock valuations that cannot be distinguished by a timed automaton with two clocks in which the absolute value of integer constants occurring in location invariants and transition guards does not exceed 2. Every dot, line segment, and white area represents some equivalence class.

Proposition 10.5.6. *Let $\mathfrak{A} := \langle \mathrm{Loc}, \ell_0, \bar{\mathrm{x}}, (\mathrm{inv}_\ell(\bar{\mathrm{x}}))_{\ell \in \mathrm{Loc}}, \mathcal{T} \rangle$ be any timed automaton and consider the transition system $\mathrm{TS}(\mathfrak{A}) := \langle S, s_0, \hookrightarrow \rangle$ associated with $\mathfrak{A}$. Given two clock valuations $\bar{r}, \bar{s} \in \mathbb{Q}_{\geq 0}^m$ with $\bar{r} \sim_{\mathfrak{A}} \bar{s}$ and any location $\ell \in \mathrm{Loc}$, we have $s_0 \hookrightarrow^* \langle \ell, \bar{r} \rangle$ if and only if $s_0 \hookrightarrow^* \langle \ell, \bar{s} \rangle$.*

Proof. This is an immediate consequence of the definition of $\mathrm{TS}(\mathfrak{A})$ and the observation that the clock constraints in $\mathfrak{A}$ cannot distinguish $\sim_{\mathfrak{A}}$-equivalent clock valuations. The latter results from Proposition 10.0.6. □

Fietzke and Weidenbach have presented an encoding of the semantic transition system of a given timed automaton $\mathfrak{A}$ into a first-order clause set with linear arithmetic constraints [FW12]. We shall use this encoding as a starting point.

Definition 10.5.7 (FOL(LA) encoding of a timed automaton, [FW12])**.**
Let $\mathfrak{A} := \langle \mathrm{Loc}, \ell_0, \bar{\mathrm{x}}, (\mathrm{inv}_\ell(\bar{\mathrm{x}}))_{\ell \in \mathrm{Loc}}, \mathcal{T} \rangle$ be a timed automaton. The FOL(LA) encoding of $\mathfrak{A}$ *is the set $N_{\mathfrak{A}}$ containing the following clauses, where* Reach *is a $(m+1)$-ary predicate symbol of sort $\mathrm{Loc} \times \mathbb{Q} \times \ldots \times \mathbb{Q}$, the $\ell \in \mathrm{Loc}$ are reused as free-sort constant symbols, $\bar{\mathrm{x}}'$ is some m-tuple of pairwise-distinct clock variables, and z is one more first-order variable (we assume $\bar{\mathrm{x}}$, $\bar{\mathrm{x}}'$, and $\{z\}$ to be pairwise disjoint):*

the initial clause	$\bigwedge_{x_i \in \bar{\mathrm{x}}} x_i = 0 \ \wedge \ \mathrm{inv}_{\ell_0}(\bar{\mathrm{x}}) \rightarrow \mathrm{Reach}(\ell_0, \bar{\mathrm{x}})$;
delay clauses	$z \geq 0 \ \wedge \ \bigwedge_{x_i \in \bar{\mathrm{x}}} x_i' = x_i + z \ \wedge \ \mathrm{inv}_{\ell}(\bar{\mathrm{x}}') \ \wedge \ \mathrm{Reach}(\ell, \bar{\mathrm{x}}) \rightarrow \mathrm{Reach}(\ell, \bar{\mathrm{x}}')$ *for every location* $\ell \in \mathrm{Loc}$;
transition clauses	$\psi(\bar{\mathrm{x}}) \ \wedge \ \bigwedge_{x_i \in Z} x_i' = 0 \ \wedge \bigwedge_{x_i \in \bar{\mathrm{x}} \setminus Z} x_i' = x_i \ \wedge \ \mathrm{inv}_{\ell'}(\bar{\mathrm{x}}') \ \wedge \ \mathrm{Reach}(\ell, \bar{\mathrm{x}}) \rightarrow \mathrm{Reach}(\ell', \bar{\mathrm{x}}')$ *for every location transition* $\langle \ell, \psi(\bar{\mathrm{x}}), Z, \ell' \rangle \in \mathcal{T}$.

Proposition 10.5.8 (Corollary 4.3 and Proposition 4.4 in [FW12])**.**
Let $\mathfrak{A} := \langle \mathrm{Loc}, \ell_0, \bar{\mathrm{x}}, (\mathrm{inv}_\ell(\bar{\mathrm{x}}))_{\ell \in \mathrm{Loc}}, \mathcal{T} \rangle$ *be a timed automaton, let* $\mathrm{TS}(\mathfrak{A}) = \langle S, s_0, \hookrightarrow \rangle$ *be its semantic transition system, and let* $N_{\mathfrak{A}}$ *be its FOL(LA) encoding.*

(i) Let $\mathcal{A}$ *be a model of* $N_{\mathfrak{A}}$*. For every location* $\ell \in \mathrm{Loc}$ *and every tuple* $\bar{r} \in \mathbb{Q}^m_{\geq 0}$*, we have* $\mathcal{A} \models \mathrm{Reach}(\ell, \bar{r})$ *if and only if* $s_0 \hookrightarrow^* \langle \ell, \bar{r} \rangle$*.*

(ii) Let $\psi(\bar{\mathrm{x}}) \in \mathrm{CC}(\bar{\mathrm{x}})$ *be some clock constraint describing the set* $S := \{\bar{r} \in \mathbb{Q}^m_{\geq 0} \mid \mathbb{Q} \models \psi(\bar{r})\}$ *of clock valuations. Moreover, let* $\ell \in \mathrm{Loc}$ *be some location in* $\mathfrak{A}$*. Any pair* $\langle \ell, \bar{r} \rangle$ *with* $\bar{r} \in S$ *is reachable in the transition system* $\mathrm{TS}(\mathfrak{A})$ *if and only if the clause set* $N_{\mathfrak{A}} \cup \{\psi(\bar{\mathrm{x}}) \wedge \mathrm{Reach}(\ell, \bar{\mathrm{x}}) \rightarrow \mathtt{false}\}$ *does not have a model.*

In the FOL(LA) encoding described in Definition 10.5.7 the passage of time is formalized in a synchronous fashion in delay clauses. This is done by adding the constraint $z \geq 0 \wedge \bigwedge_{x_i \in \bar{\mathrm{x}}} x_i' = x_i + z$ to the premise of the delay clause, where z is implicitly universally quantified (with respect to the whole clause). Since z does not occur in the rest of the delay clause, we could equivalently use the constraint $x_1 \leq x_1' \wedge \bigwedge_{x_i \in \bar{\mathrm{x}}} x_1 - x_i = x_1' - x_i'$ instead.

Next, we argue that the passage of time does not have to be formalized as a synchronous progression of all clocks. Instead, it is sufficient to require that clocks progress in such a way that their valuations do not drift apart excessively. Although this weakens the semantics slightly, reachability remains unaffected. Figure 10.4 illustrates the underlying idea. We first prove an

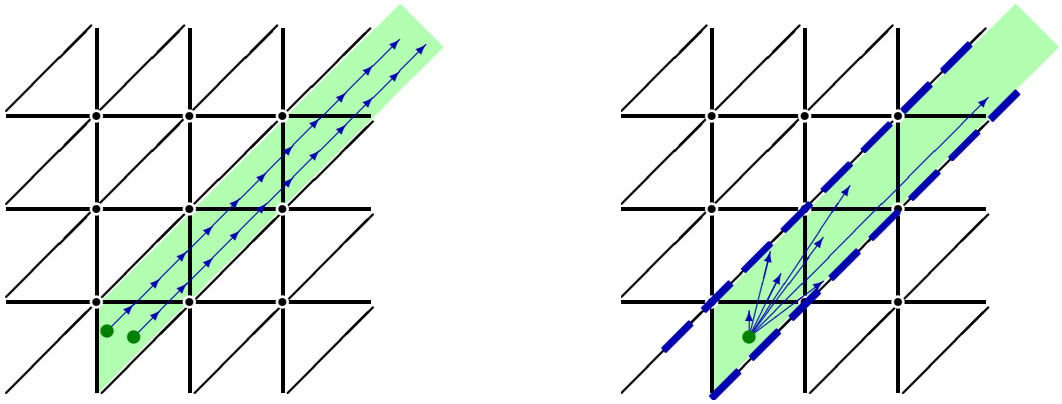


Figure 10.4: Synchronous versus asynchronous progress of time for two clocks. Left-hand side: Synchronous progress of time illustrated for two starting valuations within the same TA region. Right-hand side: Asynchronous progress of time illustrated for one starting valuation. The dashed blue lines mark the boundaries within which drifting of clocks does not affect reachability. In both cases the light green area depicts the union of TA regions that is reachable from the TA region containing the starting valuations.

auxiliary result.

Lemma 10.5.9. *Let* μ *be any positive integer and let* $S \in [0, \mu + 1)^m / \simeq_\mu$ *be any equivalence class*

T_1, T_2 *with respect to* $\simeq_\mu$. *We define the two sets* T_1, T_2 *as follows:*

$$T_1 := \big\{\bar{q}' \in [0,\mu+1)^m \;\big|\; \text{there is some } \bar{q} \in S \text{ such that for every } i,\ 1 \le i \le m,\\ \text{we have } q_i \le q'_i \text{ and } q_1 - q_i = q'_1 - q'_i\big\}\ ,$$

and

$$T_2 := \big\{\bar{q}' \in [0,\mu+1)^m \;\big|\; \\ \text{there is some } \bar{q} \in S \text{ such that for all } i_1, i_2,\ 1 \le i_1, i_2 \le m,\\ q_{i_1} \le q'_{i_1} \text{ and for every integer } k,\ -\mu \le k \le \mu, \text{ we have}\\ q_{i_1} - q_{i_2} \le k \text{ if and only if } q'_{i_1} - q'_{i_2} \le k, \text{ and}\\ q_{i_1} - q_{i_2} \ge k \text{ if and only if } q'_{i_1} - q'_{i_2} \ge k\big\}\ .$$

Then, we observe $T_1 = T_2$.

Proof. We obviously have $T_1 \subseteq T_2$.

$\bar{q}'$, $\bar{s}$ In order to prove $T_2 \subseteq T_1$, consider any $\bar{q}' \in T_2$. Pick some $\bar{s} \in S$ for which $s_i \le q'_i$ for every i. Such an $\bar{s}$ must exist because of $S \subseteq T_2$. By construction of T_2, we observe $\lfloor s_i - s_j \rfloor = \lfloor q'_i - q'_j \rfloor$ and $\lceil s_i - s_j \rceil = \lceil q'_i - q'_j \rceil$ for all i, j.

Claim I: For all indices i, j, $1 \le i, j \le m$, we have $\mathrm{fr}(s_i) = \mathrm{fr}(s_j)$ if and only if $\mathrm{fr}(q'_i) = \mathrm{fr}(q'_j)$.

Proof: For all rational numbers r, t we have $\mathrm{fr}(r) = \mathrm{fr}(t)$ if and only if $\lfloor r - t \rfloor = \lceil r - t \rceil$. Using this fact, we get that $\mathrm{fr}(s_i) = \mathrm{fr}(s_j)$ entails $\lfloor q'_i - q'_j \rfloor = \lfloor s_i - s_j \rfloor = \lceil s_i - s_j \rceil = \lceil q'_i - q'_j \rceil$ which in turn implies $\mathrm{fr}(q'_i) = \mathrm{fr}(q'_j)$. Symmetrically, $\mathrm{fr}(q'_i) = \mathrm{fr}(q'_j)$ entails $\mathrm{fr}(s_i) = \mathrm{fr}(s_j)$. ◊

Claim II: Let $k_1, \ldots, k_m$be some enumeration of the indices in $\{1, \ldots, m\}$ such that $\mathrm{fr}(s_{k_1}) \le \ldots \le \mathrm{fr}(s_{k_m})$. There is some ℓ such that

$$\mathrm{fr}(q'_{k_{\ell+1}}) \le \ldots \le \mathrm{fr}(q'_{k_m}) \le \mathrm{fr}(q'_{k_1}) \le \ldots \le \mathrm{fr}(q'_{k_\ell}).$$

Proof: Suppose Claim II does not hold, while Claim I is satisfied. More precisely, suppose there are indices j_1, j_2, j_3 such that $\mathrm{fr}(s_{j_1}) < \mathrm{fr}(s_{j_2}) < \mathrm{fr}(s_{j_3})$ and $\mathrm{fr}(q'_{j_3}) < \mathrm{fr}(q'_{j_2}) < \mathrm{fr}(q'_{j_1})$ (or $\mathrm{fr}(q'_{j_2}) < \mathrm{fr}(q'_{j_1}) < \mathrm{fr}(q'_{j_3})$ or $\mathrm{fr}(q'_{j_1}) < \mathrm{fr}(q'_{j_3}) < \mathrm{fr}(q'_{j_2})$ — these cases can be treated in an analogous fashion).

For all rational numbers r, t we have $\lfloor r - t \rfloor = \lfloor r \rfloor - \lfloor t \rfloor + \lfloor \mathrm{fr}(r) - \mathrm{fr}(t) \rfloor$, where

$$\lfloor \mathrm{fr}(r) - \mathrm{fr}(t) \rfloor = \begin{cases} 0 & \text{if } \mathrm{fr}(r) \ge \mathrm{fr}(t) \\ -1 & \text{if } \mathrm{fr}(r) < \mathrm{fr}(t). \end{cases}$$

Hence, we get the following system of equations:

$$\begin{array}{lllll} \lfloor s_{j_1} \rfloor - \lfloor s_{j_2} \rfloor - 1 &=& \lfloor s_{j_1} - s_{j_2} \rfloor &=& \lfloor q'_{j_1} - q'_{j_2} \rfloor = \lfloor q'_{j_1} \rfloor - \lfloor q'_{j_2} \rfloor \\ \lfloor s_{j_1} \rfloor - \lfloor s_{j_3} \rfloor - 1 &=& \lfloor s_{j_1} - s_{j_3} \rfloor &=& \lfloor q'_{j_1} - q'_{j_3} \rfloor = \lfloor q'_{j_1} \rfloor - \lfloor q'_{j_3} \rfloor \\ \lfloor s_{j_2} \rfloor - \lfloor s_{j_3} \rfloor - 1 &=& \lfloor s_{j_2} - s_{j_3} \rfloor &=& \lfloor q'_{j_2} - q'_{j_3} \rfloor = \lfloor q'_{j_2} \rfloor - \lfloor q'_{j_3} \rfloor \end{array}$$

As this system entails $0 = 1$, we obtain a contradiction. ◊

It remains to prove the existence of some tuple $\bar{q} \in S$ that satisfies the following requirements:

(i) $\lfloor \bar{q} \rfloor = \lfloor \bar{s} \rfloor$ and $\lceil \bar{q} \rceil = \lceil \bar{s} \rceil$.

(ii) $\lfloor q_i - q_j \rfloor = \lfloor s_i - s_j \rfloor = \lfloor q'_i - q'_j \rfloor$ and $\lceil q_i - q_j \rceil = \lceil s_i - s_j \rceil = \lceil q'_i - q'_j \rceil$ for all i, j.

(iii) $q_1 - q_i = q'_1 - q'_i$ for every i.

(iv) $q_i \le q'_i$ for every i.

The existence of such a $\bar{q}$ would immediately entail $\bar{q}' \in T_1$. Notice that Requirement (ii) is entailed by Requirement (iii) and the definition of T_2.

Consider any i with $1 \leq i \leq m$. Requirement (i) entails that $\bar{q}$ must satisfy $q_i = \lfloor s_i \rfloor + \mathrm{fr}(q_i)$. Therefore, we set $\lfloor q_j \rfloor := \lfloor s_j \rfloor$ for every j. This entails $\lfloor q_j \rfloor$

$$\lfloor q_j \rfloor \leq \lfloor q'_j \rfloor \text{ for every } j. \tag{10.4}$$

Moreover, it follows that $q_1 - q_i = \lfloor s_1 \rfloor + \mathrm{fr}(q_1) - \lfloor s_i \rfloor - \mathrm{fr}(q_i)$ and $q'_1 - q'_i = \lfloor q'_1 \rfloor + \mathrm{fr}(q'_1) - \lfloor q'_i \rfloor - \mathrm{fr}(q'_i)$. Hence, Condition (iii) requires $\lfloor s_1 \rfloor - \lfloor s_i \rfloor + \mathrm{fr}(q_1) - \mathrm{fr}(q_i) = \lfloor q'_1 \rfloor - \lfloor q'_i \rfloor + \mathrm{fr}(q'_1) - \mathrm{fr}(q'_i)$, which is equivalent to

$$\mathrm{fr}(q_1) - \mathrm{fr}(q_i) \;=\; (\lfloor q'_1 \rfloor - \lfloor q'_i \rfloor) - (\lfloor s_1 \rfloor - \lfloor s_i \rfloor) + \mathrm{fr}(q'_1) - \mathrm{fr}(q'_i)\ . \tag{10.5}$$

We distinguish several cases:

If $\bar{q}' \in S$, we simply set $\bar{q} := \bar{q}'$. Then, Requirements (i)–(iv) are satisfied.

If there is some j such that $\lfloor s_j \rfloor = \lceil s_j \rceil$, then, by Requirement (i), q_j must satisfy $\mathrm{fr}(q_j) = 0$ and, therefore, for every ℓ, $\mathrm{fr}(q_\ell)$ is determined by (10.5). Then, Conditions (i)–(iii) are satisfied. As we have

$$q_j = \lfloor q_j \rfloor \overset{(10.4)}{\leq} \lfloor q'_j \rfloor \leq q'_j,$$

Condition (iii) entails $q_\ell \leq q'_\ell$ for every ℓ. Hence, Condition (iv) is satisfied as well.

If $\mathrm{fr}(s_1) = \ldots = \mathrm{fr}(s_m)$, we observe $\lfloor q'_j - q'_\ell \rfloor = \lfloor s_j - s_\ell \rfloor = \lceil s_j - s_\ell \rceil = \lceil q'_j - q'_\ell \rceil$ for all j, ℓ. Hence, we have $\lfloor q'_1 - q'_j \rfloor = \lceil q'_1 - q'_j \rceil$, which implies $\mathrm{fr}(q'_1) = \mathrm{fr}(q'_j)$ for every j. As this entails $q'_1 - q'_j = \lfloor q'_1 - q'_j \rfloor = \lfloor s_1 - s_j \rfloor = s_1 - s_j$, Requirements (i)–(iii) are satisfied if we set $\bar{q} := \bar{s}$. Recall that we have chosen $\bar{s}$ such that $s_j \leq q'_j$ for every j. Hence, Condition (iv) is satisfied because of $\bar{q} = \bar{s}$.

If none of the above cases apply, we have $\lfloor s_i \rfloor = \lceil s_i \rceil - 1$ for every i. Moreover, there must be indices i_1, i_2 such that $\mathrm{fr}(s_{i_1}) < \mathrm{fr}(s_{i_2})$.

Let $k_1, \ldots, k_m$ be some enumeration of the indices in $\{1, \ldots, m\}$ such that k_j

$$\mathrm{fr}(s_{k_1}) \leq \ldots \leq \mathrm{fr}(s_{k_m})\ . \tag{10.6}$$

Notice that $\mathrm{fr}(s_{k_1}) < \mathrm{fr}(s_{k_m})$ holds due to $\mathrm{fr}(s_{i_1}) < \mathrm{fr}(s_{i_2})$. By Claim II, there is some ℓ such ℓ that

$$\mathrm{fr}(q'_{k_{\ell+1}}) \leq \ldots \leq \mathrm{fr}(q'_{k_m}) \leq \mathrm{fr}(q'_{k_1}) \leq \ldots \leq \mathrm{fr}(q'_{k_\ell})\ . \tag{10.7}$$

If $\ell = 0$, i.e. $\mathrm{fr}(q'_{k_1}) \leq \ldots \leq \mathrm{fr}(q'_m)$, we set $\mathrm{fr}(\bar{q}) := \mathrm{fr}(\bar{q}')$. Then, by Equation (10.4), Requirement (iv) is satisfied. Claim I together with $\mathrm{fr}(q'_{k_1}) \leq \ldots \leq \mathrm{fr}(q'_m)$ entails that we have $\lfloor \mathrm{fr}(s_1) - \mathrm{fr}(s_j) \rfloor = \lfloor \mathrm{fr}(q'_1) - \mathrm{fr}(q'_j) \rfloor$ for every j. Consequently,

$$\lfloor s_1 \rfloor - \lfloor s_j \rfloor + \lfloor \mathrm{fr}(s_1) - \mathrm{fr}(s_j) \rfloor = \lfloor s_1 - s_j \rfloor = \lfloor q'_1 - q'_j \rfloor = \lfloor q'_1 \rfloor - \lfloor q'_j \rfloor + \lfloor \mathrm{fr}(q'_1) - \mathrm{fr}(q'_j) \rfloor$$

entails $\lfloor s_1 \rfloor - \lfloor s_j \rfloor = \lfloor q'_1 \rfloor - \lfloor q'_j \rfloor$ for every j. This means we have

$$\begin{aligned} q_1 - q_j &= \lfloor q_1 \rfloor - \lfloor q_1 \rfloor + \mathrm{fr}(q_j) - \mathrm{fr}(q_j) \\ &= \lfloor s_1 \rfloor - \lfloor s_j \rfloor + \mathrm{fr}(q'_1) - \mathrm{fr}(q'_j) \\ &= \lfloor q'_1 \rfloor - \lfloor q'_j \rfloor + \mathrm{fr}(q'_1) - \mathrm{fr}(q'_j) \\ &= q'_1 - q'_j \end{aligned}$$

for every j. In other words, Requirements (i)–(iv) are met.

If $\ell > 0$, then Claim I together with $\mathrm{fr}(s_{k_1}) < \mathrm{fr}(s_{k_m})$ entails that $\mathrm{fr}(q'_{k_m})$ is strictly smaller than $\mathrm{fr}(q'_{k_1})$. In this case, we define a rational number $\varepsilon := \frac{1}{2}\big(\mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_m})\big)$. For ε every $j \leq \ell$ we set $\mathrm{fr}(q_{k_j}) := \varepsilon + \big(\mathrm{fr}(q'_{k_j}) - \mathrm{fr}(q'_{k_1})\big)$. For every $j \geq \ell + 1$ we set $\mathrm{fr}(q_{k_j}) := \mathrm{fr}(q_{k_j})$ $\varepsilon + 1 - \big(\mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_j})\big)$.

Claim III: We get $0 < \mathrm{fr}(q_{k_1}) \leq \ldots \leq \mathrm{fr}(q_{k_\ell}) \leq \mathrm{fr}(q_{k_{\ell+1}}) \leq \ldots \leq \mathrm{fr}(q_{k_m}) < 1$.

Proof:

We observe the following:

- $\mathrm{fr}(q_{k_1}) = \varepsilon + \big(\mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_1})\big) = \varepsilon > 0$.
- $\mathrm{fr}(q_{k_m}) = \varepsilon + 1 - \big(\mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_m})\big) = \frac{1}{2}\big(\mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_m})\big) + 1 - \big(\mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_m})\big) = 1 - \frac{1}{2}\big(\mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_m})\big) < 1$.
- Because of $\mathrm{fr}(q'_{k_\ell}) \in [0,1)$ and $\mathrm{fr}(q'_{k_{\ell+1}}) \in [0,1)$, we obtain $\mathrm{fr}(q'_{k_\ell}) \leq \mathrm{fr}(q'_{k_{\ell+1}}) + 1$. Hence, we get $\mathrm{fr}(q_{k_\ell}) = \varepsilon + \mathrm{fr}(q'_{k_\ell}) - \mathrm{fr}(q'_{k_1}) \leq \varepsilon + \mathrm{fr}(q'_{k_{\ell+1}}) + 1 - \mathrm{fr}(q'_{k_1}) = \varepsilon + 1 - \big(\mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_{\ell+1}})\big) = \mathrm{fr}(q_{k_{\ell+1}})$.

The above observations entail $0 < \mathrm{fr}(q_{k_1})$, $\mathrm{fr}(q_{k_\ell}) \leq \mathrm{fr}(q_{k_{\ell+1}})$, and $\mathrm{fr}(q_{k_m}) < 1$. By definition of the $\mathrm{fr}(q_{k_j})$ and our assumptions $\mathrm{fr}(q'_{k_1}) \leq \ldots \leq \mathrm{fr}(q'_{k_\ell})$ and $\mathrm{fr}(q'_{k_{\ell+1}}) \leq \ldots \leq \mathrm{fr}(q'_{k_m})$, these observations imply Claim III. ◊

Claim IV: For every j we have $\big(\lfloor s_{k_1} \rfloor + \mathrm{fr}(q_{k_1})\big) - \big(\lfloor s_{k_j} \rfloor + \mathrm{fr}(q_{k_j})\big) = q'_{k_1} - q'_{k_j}$.

Proof: If $1 \leq j \leq \ell$, then we have

$$\begin{aligned}
&\big(\lfloor s_{k_1} \rfloor + \mathrm{fr}(q_{k_1})\big) - \big(\lfloor s_{k_j} \rfloor + \mathrm{fr}(q_{k_j})\big) \\
&= \lfloor s_{k_1} \rfloor + \varepsilon + \big(\mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_1})\big) - \lfloor s_{k_j} \rfloor - \varepsilon - \big(\mathrm{fr}(q'_{k_j}) - \mathrm{fr}(q'_{k_1})\big) \\
&= \lfloor s_{k_1} \rfloor - \lfloor s_{k_j} \rfloor + \mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_j}) \\
&= \lfloor s_{k_1} - s_{k_j} \rfloor + \delta + \mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_j}) \\
&= \lfloor q'_{k_1} - q'_{k_j} \rfloor + \delta + \mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_j}) \\
&= \lfloor q'_{k_1} \rfloor - \lfloor q'_{k_j} \rfloor + \mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_j}) \\
&= q'_{k_1} - q'_{k_j} \ ,
\end{aligned}$$

where

$$\delta := \begin{cases} 0 & \text{if } \mathrm{fr}(q'_{k_1}) = \mathrm{fr}(q'_{k_j}) \text{ or } \mathrm{fr}(s_{k_1}) = \mathrm{fr}(s_{k_j}), \text{ and} \\ 1 & \text{if } \mathrm{fr}(q'_{k_1}) < \mathrm{fr}(q'_{k_j}) \text{ or } \mathrm{fr}(s_{k_1}) < \mathrm{fr}(s_{k_j}) \end{cases}$$

(recall that we have $\mathrm{fr}(q'_{k_1}) \leq \mathrm{fr}(q'_{k_j})$ and $\mathrm{fr}(s_{k_1}) \leq \mathrm{fr}(s_{k_j})$ and that Claim I entails that $\mathrm{fr}(q'_{k_1}) = \mathrm{fr}(q'_{k_j})$ if and only if $\mathrm{fr}(s_{k_1}) = \mathrm{fr}(s_{k_j})$, hence, δ is well defined).
If $\ell + 1 \leq j \leq m$, then we have

$$\begin{aligned}
&\big(\lfloor s_{k_1} \rfloor + \mathrm{fr}(q_{k_1})\big) - \big(\lfloor s_{k_j} \rfloor + \mathrm{fr}(q_{k_j})\big) \\
&= \lfloor s_{k_1} \rfloor + \varepsilon + \big(\mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_1})\big) - \lfloor s_{k_j} \rfloor - \varepsilon - 1 + \big(\mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_j})\big) \\
&= \lfloor s_{k_1} \rfloor - \lfloor s_{k_j} \rfloor + \mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_j}) - 1 \ .
\end{aligned}$$

Since $\mathrm{fr}(q'_{k_m})$ is strictly smaller than $\mathrm{fr}(q'_{k_1})$, we get $\mathrm{fr}(q'_{k_j}) < \mathrm{fr}(q'_{k_1})$. Moreover, Claim I together with $\mathrm{fr}(s_{k_1}) \leq \mathrm{fr}(s_{k_j})$ entails $\mathrm{fr}(s_{k_1}) < \mathrm{fr}(s_{k_j})$. Hence, $\lfloor s_{k_1} - s_{k_j} \rfloor = \lfloor s_{k_1} \rfloor - \lfloor s_{k_j} \rfloor - 1$ and $\lfloor q'_{k_1} - q'_{k_j} \rfloor = \lfloor q'_{k_1} \rfloor - \lfloor q'_{k_j} \rfloor$. Consequently, we get

$$\begin{aligned}
&\lfloor s_{k_1} \rfloor - \lfloor s_{k_j} \rfloor + \mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_j}) - 1 \\
&= \lfloor s_{k_1} - s_{k_j} \rfloor + 1 + \mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_j}) - 1 \\
&= \lfloor q'_{k_1} - q'_{k_j} \rfloor + \mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_j}) \\
&= \lfloor q'_{k_1} \rfloor - \lfloor q'_{k_j} \rfloor + \mathrm{fr}(q'_{k_1}) - \mathrm{fr}(q'_{k_j}) \\
&= q'_{k_1} - q'_{k_j}.
\end{aligned}$$

◊

Claim V: For every j we have $\lfloor s_{k_j} \rfloor + \mathrm{fr}(q_{k_j}) \leq q'_{k_j}$.

Proof: As we assume $\bar{q}' \notin S$, there is at least one i such that $\lfloor q'_{k_i} \rfloor > \lfloor s_{k_i} \rfloor$. This entails $q'_{k_i} \geq \lfloor q'_{k_i} \rfloor > \lfloor s_{k_i} \rfloor + \mathrm{fr}(q_{k_i})$. One consequence of Claim IV is that we have $\big(\lfloor s_{k_i} \rfloor + \mathrm{fr}(q_{k_i})\big) - \big(\lfloor s_{k_j} \rfloor + \mathrm{fr}(q_{k_j})\big) = q'_{k_i} - q'_{k_j}$ for every j. This can be rewritten into the equivalent equation $q'_{k_j} - \big(\lfloor s_{k_j} \rfloor + \mathrm{fr}(q_{k_j})\big) = q'_{k_i} - \big(\lfloor s_{k_i} \rfloor + \mathrm{fr}(q_{k_i})\big)$. Combined with $q'_{k_i} > \lfloor s_{k_i} \rfloor + \mathrm{fr}(q_{k_i})$, this entails $q'_{k_j} > \lfloor s_{k_j} \rfloor + \mathrm{fr}(q_{k_j})$ for every j. ◊

This means, if we set $q_{k_j} := \lfloor s_{k_j} \rfloor + \mathrm{fr}(q_{k_j})$ for every j, then Requirements (i)–(iv) are satisfied. q_{k_j}

□

Now we have all necessary tools at hand to show that, if we are only interested in reachability analysis for timed automata, progress of time does not need to be synchronized over all clocks. Much rather is it sufficient to formalize the requirement that time progress is never negative and that clocks must not drift apart excessively.

Lemma 10.5.10. *Consider any delay clause*

$$C(\bar{\mathrm{x}}, \bar{\mathrm{x}}', z) := \quad z \geq 0 \ \wedge \ \textstyle\bigwedge_{x_i \in \bar{\mathrm{x}}} x'_i = x_i + z \ \wedge \ \mathrm{inv}_\ell(\bar{\mathrm{x}}') \ \wedge \ \mathrm{Reach}(\ell, \bar{\mathrm{x}}) \rightarrow \mathrm{Reach}(\ell, \bar{\mathrm{x}}')$$

that belongs to the FOL(LA) encoding of any timed automaton $\mathfrak{A} := \big\langle \mathrm{Loc}, \ell_0, \bar{\mathrm{x}}, \big(\mathrm{inv}_\ell(\bar{\mathrm{x}})\big)_{\ell \in \mathrm{Loc}}, \mathcal{T} \big\rangle$. *Let* μ *be any positive integer. Let* $M(\bar{\mathrm{x}}, \bar{\mathrm{x}}')$ *be a finite clause set corresponding to the following formula*

$$\varphi(\bar{\mathrm{x}}, \bar{\mathrm{x}}') := \Big(\bigwedge_{x_i, x_j \in \bar{\mathrm{x}}} \bigwedge_{-\mu \leq k \leq \mu} \big(x_i - x_j \leq k \ \leftrightarrow \ x'_i - x'_j \leq k\big) \ \wedge \ \big(x_i - x_j \geq k \ \leftrightarrow \ x'_i - x'_j \geq k\big)$$
$$\wedge \ \bigwedge_{x_i \in \bar{\mathrm{x}}} x'_i \geq x_i \ \wedge \ \mathrm{inv}_\ell(\bar{\mathrm{x}}') \ \wedge \ \mathrm{Reach}(\ell, \bar{\mathrm{x}}) \Big) \longrightarrow \mathrm{Reach}(\ell, \bar{\mathrm{x}}') \ .$$

For every $\simeq_\mu$*-uniform structure* $\mathcal{A}$ *we have* $\mathcal{A} \models \forall z.\, C(\bar{r}, \bar{r}', z)$ *for all tuples* $\bar{r}, \bar{r}' \in [0, \mu+1)^m$ *if and only if* $\mathcal{A} \models M(\bar{q}, \bar{q}')$ *holds for all tuples* $\bar{q}, \bar{q}' \in [0, \mu+1)^m$.

Proof. We first show that the clause C is equivalent to the clause

$$C'(\bar{\mathrm{x}}, \bar{\mathrm{x}}') := \bigwedge_{x_i \in \bar{\mathrm{x}}} x_i \leq x'_i \ \wedge \ \bigwedge_{x_i \in \bar{\mathrm{x}}} x_1 - x_i = x'_1 - x'_i \ \wedge \ \mathrm{inv}_\ell(\bar{\mathrm{x}}') \ \wedge \ \mathrm{Reach}(\ell, \bar{\mathrm{x}}) \rightarrow \mathrm{Reach}(\ell, \bar{\mathrm{x}}') \ .$$

Since the variable z in C occurs only in the premise, $\forall z.\, C(\bar{\mathrm{x}}, \bar{\mathrm{x}}', z)$ is equivalent to

$$(\exists z.\ z \geq 0 \wedge \bigwedge_{x_i \in \bar{\mathrm{x}}} x'_i - x_i = z) \wedge \mathrm{inv}_\ell(\bar{\mathrm{x}}') \wedge \mathrm{Reach}(\ell, \bar{\mathrm{x}}) \rightarrow \mathrm{Reach}(\ell, \bar{\mathrm{x}}') \ .$$

For the part $\exists z.\ z \geq 0 \wedge \bigwedge_{x_i \in \bar{\mathrm{x}}} x'_i - x_i = z$ in the latter formula we observe

$$\begin{aligned}
&\exists z.\ z \geq 0 \ \wedge \bigwedge_{x_i \in \bar{\mathrm{x}}} x'_i - x_i = z \\
\models\!\!\models \ &\bigwedge_{x_i \in \bar{\mathrm{x}}} x_i \leq x'_i \ \wedge \bigwedge_{x_i, x_j \in \bar{\mathrm{x}}} x'_i - x_i = x'_j - x_j \\
\models\!\!\models \ &\bigwedge_{x_i \in \bar{\mathrm{x}}} x_i \leq x'_i \ \wedge \bigwedge_{x_i \in \bar{\mathrm{x}}} x'_1 - x_1 = x'_i - x_i \\
\models\!\!\models \ &\bigwedge_{x_i \in \bar{\mathrm{x}}} x_i \leq x'_i \ \wedge \bigwedge_{x_i \in \bar{\mathrm{x}}} x_1 - x_i = x'_1 - x'_i \ .
\end{aligned}$$

Consequently, the formulas $\forall z.\, C(\bar{\mathrm{x}}, \bar{\mathrm{x}}', z)$ and $C'(\bar{\mathrm{x}}, \bar{\mathrm{x}}')$ are equivalent.

Let $\mathcal{A}$ be any $\simeq_\mu$-uniform structure.

Claim I: For every equivalence class $S \subseteq [0, \mu+1)^m$ with respect to $\simeq_\mu$ we have that, if $\mathcal{A} \models \mathrm{Reach}(\ell, \bar{r})$ holds for one $\bar{r} \in S$, then $\mathcal{A} \models \mathrm{Reach}(\ell, \bar{q})$ holds for every $\bar{q} \in S$.

Proof: By $\simeq_\mu$-uniformity of $\mathcal{A}$.

Now suppose $\mathcal{A} \models C'(\bar{r}, \bar{r}')$ holds for all tuples $\bar{r}, \bar{r}' \in [0, \mu+1)^m$. Moreover, suppose there is some pair of tuples $\bar{q}, \bar{q}' \in [0, \mu+1)^m$ such that $\mathcal{A} \not\models \varphi(\bar{q}, \bar{q}')$. In other words, $\mathcal{A}, [\bar{\mathrm{x}} \mapsto \bar{q}, \bar{\mathrm{x}}' \mapsto \bar{q}']$ satisfies the premise of φ — among them $\mathrm{inv}_\ell(\bar{\mathrm{x}}')$ — but does not satisfy the consequent $\mathrm{Reach}(\ell, \bar{\mathrm{x}}')$.

S Let $S := [\bar{q}]_{\simeq_\mu}$ be the equivalence class with respect to $\simeq_\mu$ to which $\bar{q}$ belongs. As we have

$$\mathcal{A} \models \bigwedge_{x_i, x_j \in \bar{\mathrm{x}}} \bigwedge_{-\mu \leq k \leq \mu} \big(q_i - q_j \leq k \;\leftrightarrow\; q'_i - q'_j \leq k\big) \;\wedge\; \big(q_i - q_j \geq k \leftrightarrow q'_i - q'_j \geq k\big) \;\wedge\; \bigwedge_{x_i \in \bar{\mathrm{x}}} q'_i \geq q_i \;,$$

T_2 we conclude that $\bar{q}' \in T_2$, where T_2 is defined like in Lemma 10.5.9, based on S. Moreover, we know that $\mathcal{A} \models \mathrm{Reach}(\ell, \bar{s})$ for every $\bar{s} \in S$, as $\mathcal{A}$ is $\simeq_\mu$-uniform. The fact that $\mathcal{A} \models C'(\bar{r}, \bar{r}')$ holds for all tuples $\bar{r}, \bar{r}' \in [0, \mu+1)^m$ entails $\mathcal{A} \models \mathrm{Reach}(\ell, \bar{s}')$ for every $\bar{s}' \in T_1$ for which $\mathbb{Q} \models \mathrm{inv}_\ell(\bar{s}')$, where T_1 T_1 is defined like in Lemma 10.5.9, based on S. Hence, Lemma 10.5.9 entails $\mathcal{A} \models \mathrm{Reach}(\ell, \bar{s}'')$ for every $\bar{s}'' \in T_2$ for which $\mathbb{Q} \models \mathrm{inv}_\ell(\bar{s}'')$, in particular for $\bar{s}'' = \bar{q}'$. This contradiction implies that $\mathcal{A} \models \varphi(\bar{q}, \bar{q}')$ holds for all tuples $\bar{q}, \bar{q}' \in [0, \mu+1)^m$.

The opposite direction can be argued analogously. Consequently, we have $\mathcal{A} \models \forall z.\, C(\bar{r}, \bar{r}', z)$ for all tuples $\bar{r}, \bar{r}' \in [0, \mu+1)^m$ if and only if $\mathcal{A} \models \varphi(\bar{q}, \bar{q}')$ holds for all tuples $\bar{q}, \bar{q}' \in [0, \mu+1)^m$. □

$\mathfrak{A}$, κ, μ Consider any timed automaton $\mathfrak{A} := \big\langle \mathrm{Loc}, \ell_0, \bar{\mathrm{x}}, \big(\mathrm{inv}_\ell(\bar{\mathrm{x}})\big)_{\ell \in \mathrm{Loc}}, \mathcal{T} \big\rangle$, let κ be the smallest positive integer that is larger than the absolute value of any integer occurring in any clock constraint in $\mathfrak{A}$, and let $\mu := \kappa \cdot m$ (recall that $m = |\bar{\mathrm{x}}|$). In order to decide for $\mathfrak{A}$ which states $\langle \ell, \bar{r} \rangle$ are reachable, Proposition 10.5.6 entails that it is sufficient to consider a bounded subspace of $\mathbb{Q}^m$. More precisely, any valuation $\bar{r} \in \mathbb{Q}^m_{\geq 0}$ of $\mathfrak{A}$'s clocks can be projected to some $\sim_{\mathfrak{A}}$-equivalent valuation $\bar{r}' \in [0, \mu+1)^m$ that $\mathfrak{A}$ cannot distinguish from $\bar{r}$. In the subspace $[0, \mu+1)^m$, $\mathfrak{A}$'s TA regions coincide with (finite unions of) equivalence classes with respect to $\simeq_\mu$ (cf. Definition 10.4.1). In fact, the quotient $[0, \mu+1)^m/_{\simeq_\mu}$, i.e. the partition of $[0, \mu+1)^m$ into finitely many equivalence classes induced by $\simeq_\mu$, constitutes a refinement of the division of $[0, \mu+1)^m$ into TA regions with respect to $\sim_{\mathfrak{A}}$. That is, for every set $S \in [0, \mu+1)^m/_{\simeq_\mu}$ there is some set $T \in [0, \mu+1)^m/_{\sim_{\mathfrak{A}}}$ such that $S \subseteq T$. Conversely, for every set $T \in [0, \mu+1)^m/_{\sim_{\mathfrak{A}}}$ there is a finite collection of sets $S_1, \ldots, S_k \in [0, \mu+1)^m/_{\simeq_\mu}$ such that $T = S_1 \cup \ldots \cup S_k$. Since, by Proposition 10.5.6, every pair $\langle \ell, \bar{r} \rangle$ with $\bar{r} \in R$ for some TA region $R \in \mathbb{Q}^m_{\geq 0}/_{\sim_{\mathfrak{A}}}$ is reachable if and only if all pairs $\langle \ell, \bar{r}' \rangle$ with $\bar{r}' \in R$ are reachable, any *minimal* model $\mathcal{A}$ of the FOL(LA) encoding $N_{\mathfrak{A}}$ is $\simeq_\mu$-uniform (where minimality of $\mathcal{A}$ refers to the minimality of the set $\mathrm{Reach}^{\mathcal{A}}$ with respect to set inclusion $\subseteq$). This is why it is sufficient that Lemma 10.5.10 focuses on $\simeq_\mu$-uniform structures.

Given the FOL(LA) encoding $N_{\mathfrak{A}}$ of $\mathfrak{A}$, we obtain a BSR(BD) encoding $N'_{\mathfrak{A}}$ of reachability with respect to $\mathfrak{A}$ in the following two steps:

(1) Replace every delay clause in $N_{\mathfrak{A}}(\bar{\mathrm{x}}, \bar{\mathrm{x}}', z)$ with the clauses from the finite set $M(\bar{\mathrm{x}}, \bar{\mathrm{x}}')$ constructed in Lemma 10.5.10, where we use $\mu := \kappa \cdot m$.

(2) Conjoin the formula $0 \leq y \;\wedge\; y < \mu + 1$ to every Λ-part of clauses in which a base-sort variable y occurs.

Since any $\widehat{\simeq}_{\mu+1}$-uniform model of $N'_{\mathfrak{A}}$ is $\simeq_\mu$-uniform over the subspace $(-\mu-1, \mu+1)^m$, Lemma 10.5.10 entails that $N'_{\mathfrak{A}}$ faithfully encodes reachability for $\mathfrak{A}$.

Theorem 10.5.11. *The reachability problem for a given timed automaton can be expressed in terms of satisfiability of a finite BSR(BD) clause set.*

Chapter 11

Undecidable Fragments of Linear Arithmetic with Uninterpreted Predicate Symbols

In Chapter 8 we have already discussed that adding uninterpreted predicate symbols to the language of Presburger arithmetic renders the associated satisfiability problem undecidable. Indeed, already the availability of a single uninterpreted unary predicate symbol P — recall that we have baptized this language PA+P in Chapter 9 — results in a satisfiability problem that is not even semi-decidable. The latter was observed by Halpern in 1991 [Hal91]. Halpern's proof rests on a result by Harel, Pnueli, and Stavi (Proposition 5.1 in [HPS83]), which states that the set of *Gödel numbers* of recurring Turing machines is Σ^1_1-complete.[1] A nondeterministic Turing machine is considered to be *recurring* if, started on an empty input tape, it is able to perform a nonterminating computation in which it infinitely often reaches its initial state (but not necessarily its initial configuration). The encoding of recurring Turing machines that Halpern employs in his proof results in formulas with two quantifier alternations. More precisely, the used sentences start with a $\forall^*\exists^*\forall^*$-prefix of first-order quantifiers when written in prenex normal form. However, this pattern of quantifier alternations can be simplified to $\forall^*\exists^*$, as pointed out by Speranski in [Spe13a]. Formally, Downey's encoding of two-counter machines in [Dow72] exhibits a $\forall\exists$ alternation as well. However, suitable modifications lead to an encoding that does not require existential quantification at all, see Section 11.2.4.

recurring Turing machine

In the present chapter we develop refined undecidability results that restrict the used language even further, e.g. by considering only the universal subfragment, or by allowing only very simple arithmetic atoms. Most of the presented results will be based on a novel encoding of the runs of two-counter machines that we shall reuse multiple times in slightly different variants. A crucial difference between Downey's encoding and ours is that the former concentrates on reachability of configurations, while the latter also considers the temporal order in which configurations are reached. One consequence is that our encoding facilitates the formalization of *recurrence* for nondeterministic two-counter machines. This requires some chronological information regarding the configurations that occur in a run, which goes beyond reachability.

In Section 11.1 we will give some basic definitions and first undecidability results based on fairly simple and straightforward encodings of two-counter machines. More complicated encodings under stronger syntatic restrictions will follow in the subsequent sections. In Sections 11.1 and 11.2 we shall restrict the admitted language so that only universal first-order quantifiers may be used (in prenex sentences). Yet, the associated validity and satisfiability problems remain undecidable. To

[1] Halpern's proof shifts the perspective from the validity problem to the problem of satisfiability. A Σ^1_1-complete satisfiability problem entails a Π^1_1-complete validity problem and vice versa, given that the considered languages are closed under negation. For the definition of the analytical hierarchy and the sets Π^1_1 and Σ^1_1, see, e.g., Chapter IV.2 in [Odi92] or Chapter 16 in [Rog87].

be more precise, we show Σ_1^0-completeness of the set of unsatisfiable sentences from the universal fragment of PA+P (cf. Theorems 11.2.2 and 11.3.3). As it turns out, this result is still valid when we use the rationals or reals as the underlying domain (Theorem 11.2.6). Our proof proceeds by a reduction of the (negated) halting problem for two-counter machines (cf. [Min67]) to the satisfiability problem in the described language. A run of such a machine, started with a certain input, can be represented by a potentially infinite sequence of configurations $\langle \ell, c_1, c_2 \rangle$ — triples of natural numbers —, where ℓ denotes the current control state of the machine and c_1, c_2 are the current values of the machine's counters. It is not very hard to imagine that such a sequence of configurations can be encoded by a potentially infinite sequence of bits. On the other hand, we can conceive any interpretation of a unary predicate P over the natural numbers as a bit sequence. Given this basic idea, it remains to devise a translation of the program of an arbitrary two-counter machine into a suitable sentence from the universal fragment of PA+P. Suitable in this case means that any model of the resulting sentence interprets P such that it faithfully represents a run of the given machine on the given input.

In Section 11.3 we will relax our language restrictions a bit and show that allowing one quantifier alternation entails a high degree of undecidability. More precisely, the set of satisfiable $\forall^*\exists$-$\Sigma_{\mathrm{PA}+P}$-sentences is Σ_1^1-complete. The proof rests on a lemma that is due to Alur and Henzinger [AH94] and that rephrases Harel et al.'s Σ_1^1-hardness result for recurring Turing machines in terms of recurring two-counter machines. In order to apply this lemma, we will have to adapt the encoding presented in Section 11.2 only slightly. All we need to do is to add the possibility of nondeterministic branching of the control flow and to replace the check for the reachability of the `halt` instruction by a condition that formalizes the recurrence property. Moreover, we will observe that our undecidability and Σ_1^1-hardness results in the integer setting can be transfered to corresponding results in the realm of rational and real numbers. We will do so at the end of Sections 11.2 and 11.3, respectively.

In Section 11.4 we shall develop an encoding of two-counter machines that only uses difference constraints and where the interpretation of the occurring uninterpreted predicate symbols can be restricted to finite subsets of the rational interval $[0, 1]$. In the presence of a $\exists\forall$ quantifier alternation, the associated satisfiability problem is undecidable. This nicely contrasts our findings from Section 10.4, where we have shown that satisfiability of finite BSR(BD) clause sets is decidable.

Finally, we will discuss the relevance of our findings to the field of verification in Section 11.5. In particular, we will derive undecidability results for quantified fragments of separation logic (Section 11.5.1), the theory of arrays (Section 11.5.2), and combinations of the theory of equality over uninterpreted functions with restricted forms of integer arithmetic (Sections 11.5.3 and 11.5.4). In certain cases, our results even imply the absence of sound and complete deductive calculi.

It should be stressed once again that most of the results outlined above are obtained based on refinements of the encoding of two-counter machines presented in Section 11.2. To the author's knowledge, a similarly general applicability is not documented for any other encoding of undecidable problems in the language of Presburger arithmetic augmented with uninterpreted predicate symbols.

11.1 Minsky's Two-Counter Machines, the Universal fragment of Presburger Arithmetic, and Simple Encodings

In Chapter 9 we have defined the language of *Presburger arithmetic* to comprise all first-order formulas with equality over the vocabulary $\Sigma_{\mathrm{PA}} = \langle \{<, \leq, =, \neq\}, \{0, 1, +, -\} \rangle$, where the only sort is $\mathbb{Z}$. The *universal fragment of Presburger arithmetic* confines the language of Presburger arithmetic to sentences in prenex normal form in which only universal quantification is allowed and existential quantification may not occur. Analogously, we say that the *universal fragment of* PA+P is the set of all prenex $\Sigma_{\mathrm{PA}+P}$-sentences without existential quantifiers, where $\Sigma_{\mathrm{PA}+P}$ is the vocabulary $\langle \{<, \leq, =, \neq, P\}, \{0, 1, +, -\} \rangle$ with the uninterpreted unary predicate symbol P of sort $\mathbb{Z}$, as we have defined it in Chapter 9.

universal Presburger arithmetic

universal fragment of PA+P

Minsky has introduced the two-counter machine as a Turing-complete model of computation

(Theorem 14.1-1 in [Min67]). We shall only briefly recap the basic architecture of this kind of computing device.

Definition 11.1.1. *A* two-counter machine $\mathcal{M}$ *consists of two counters* C_1, C_2 *and a finite program whose lines are labeled with integers* $0, \ldots, K$ *for some nonnegative integer* K*. Each program line contains one of five possible instructions with the following meaning:*

`inc`(C_1)	*increment counter* C_1 *and proceed with the next instruction;*
`inc`(C_2)	*increment counter* C_2 *and proceed with the next instruction;*
`test&dec`(C_1,ℓ)	*if* $C_1 > 0$ *then decrement* C_1 *and proceed with the next instruction, otherwise proceed with instruction* ℓ *and leave the counters unchanged;*
`test&dec`(C_2,ℓ)	*if* $C_2 > 0$ *then decrement* C_2 *and proceed with the next instruction, otherwise proceed with instruction* ℓ *and leave the counters unchanged;*
`goto`(ℓ)	*leave the counters unchanged and proceed with instruction* ℓ*;*
`halt`	*halt the computation.*

We tacitly assume that the last program line, i.e. line K*, of any two-counter machine contains the* `halt` *instruction and that there is no other line containing* `halt`*. This assumption is not a restriction, as the* `goto` *instruction is available. In the initial state of a given two-counter machine the input is stored in the two counters. The computation of the machine starts at the first program line, labeled* 0*. We occasionally refer to the initial and last program line as* ℓ_{init} *and* ℓ_{halt}*,* ℓ_{init}*,* ℓ_{halt} *respectively.*

A run *of a two-counter machine* $\mathcal{M}$ *is a possibly infinite sequence of triples* $\langle \ell, c_1, c_2 \rangle$*, also called* configurations*, with three nonnegative integer values each, where* ℓ *denotes the current program line — to be executed in the next step — and* c_1, c_2 *denote the current values of the two counters* C_1, C_2*, respectively. The first triple in a run has the form* $\langle 0, m, n \rangle$*, where* m, n *constitutes the input. Given two successive triples* $\langle \ell, c_1, c_2 \rangle$ *and* $\langle \ell', c_1', c_2' \rangle$ *in a run, the latter is the result of applying program line* ℓ *to the configuration* $\langle \ell, c_1, c_2 \rangle$*. For instance, the successor of* $\langle \ell, c_1, c_2 \rangle$ *is* $\langle \ell + 1, c_1, c_2 + 1 \rangle$ *if the* ℓ*-th program line of* $\mathcal{M}$ *is* `inc`(C_2)*. Only finite runs contain a triple* $\langle K, c_1, c_2 \rangle$ *where* K *is the program line containing the* `halt` *instruction, and then* $\langle K, c_1, c_2 \rangle$ *is the very last triple in the sequence.*

Notice that the described machine model describes deterministic computation processes. Since the machine model is strong enough to simulate any deterministic Turing machine, the halting problem for two-counter machines is undecidable.

Proposition 11.1.2 (Corollary of Theorem 14.1-1 from [Min67])**.** *It is impossible to devise an algorithm that is able to decide for every two-counter machine* $\mathcal{M}$ *and every input* $\langle m, n \rangle \in \mathbb{N} \times \mathbb{N}$ *whether* $\mathcal{M}$ *ever reaches a program line containing the* `halt` *instruction when started on* $\langle m, n \rangle$*.*

It turns out that two-counter machines can be encoded easily even when only a very restricted syntax is allowed for arithmetic atoms. In what follows we shall take a look at several simple encodings where uninterpreted predicate symbols of arity greater than one are used and the arithmetic atoms are restricted to one of four categories: (1) *difference constraints* $x - y \lhd c$, (2) *additive constraints* $x + y \lhd c$, (3) *quotient constraints* $x \lhd c \cdot y$ (which could equivalently be written $\frac{x}{y} \lhd c$, hence the name), and (4) *multiplicative constraints*[2] In case of quotient and multiplicative constraints one could also use the rational or integer domain and formulate an encoding in such a way that imposing lower and upper bounds on the used variables does not result in a decidable fragment — which would be the case if we were using variables over the integers. We shall devise such an encoding below and, based on a fragment only containing difference constraints, in Section 11.4.

[2]While atoms of the form $x \lhd c \cdot y$, with c being any nonnegative integer, can be read as an abbreviation of PA terms $x \lhd \underbrace{y + \ldots + y}_{c \text{ times}}$, atoms of the form $x \cdot y \lhd c$ cannot. We view the latter as *nonlinear arithmetic terms* either over the integers or over the reals. They will not play any significant role in the rest of the present thesis.

We start with difference constraints. We use the predicate symbol $M : \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$ to address the state of the two-counter machine as follows: $M(u, x, y, z)$ stands for a machine at instruction u with counter values $c_1 = x - z - 1$ and $c_2 = y - z - 1$, where the last argument z keeps track of an offset relative to which x and y store the values of the counters. Following this principle, the increment instruction for the first counter C_1 is realized by adding 1 to the x-component of $M(u, x, y, z)$ and leaving the offset z untouched. Decrementing the first counter then amounts to adding 1 to the y- and z-component of $M(u, x, y, z)$, i.e. relative to the offset stored in the z-component the value of the x-component is reduced by one, while the value in the y-component keeps its distance to the offset in z. Moreover, we use the relative distance of 1 between a counter and the offset to encode zero. Hence, the initial state is set to $M(\ell_{\text{init}}, 1, 1, 0)$, i.e. the offset (last argument) starts at 0, while the second and third components start at 1, which is intended to represent the counter values zero. In this way, we make sure that it is sufficient to use only the constant 1 in all the difference constraints. The offset is an appropriate tool that allows us to have a uniform syntactic structure for all atomic constraints. It is due to the offset encoding that we can easily use a difference constraint when checking whether a counter is zero or not.

In Table 11.1 we give prototypical encodings of the instructions of two-counter machines. The encoding is only given for counter C_1; the encoding for counter C_2 can be done analogously. The symbols $\ell, \ell', \ell_{\text{init}}, \ell_{\text{halt}}$ are placeholders for concrete nonnegative integers that are used as labels for program statements. Whenever we write $\ell + 1$, we mean the natural number that is the result of incrementing ℓ and not the formal term "$\ell + 1$". The symbols m, n stand for concrete integers that constitute the input.

Operation	Encoding
$\ell : \texttt{inc}(C_1)$	$\forall xyzx'.\ x' - x = 1\ \wedge\ M(\ell, x, y, z) \rightarrow M(\ell + 1, x', y, z)$
$\ell : \texttt{test\&dec}(C_1, \ell')$	$\big(\forall xyzx'y'z'.\ x - z > 1 \wedge y' - y = 1 \wedge z' - z = 1\ \wedge\ M(\ell, x, y, z) \rightarrow M(\ell + 1, x, y', z')\big)$
	$\wedge\ \big(\forall xyz.\ x - z = 1\ \wedge\ M(\ell, x, y, z) \rightarrow M(\ell', x, y, z)\big)$
$\ell : \texttt{goto}(\ell')$	$\forall xyz.\ M(\ell, x, y, z) \rightarrow M(\ell', x, y, z)$
$\ell_{\text{halt}} : \texttt{halt}$	$\forall xyz.\ M(\ell_{\text{halt}}, x, y, z) \rightarrow M(\ell_{\text{halt}}, 0, 0, 0)$
Initial condition:	$M(\ell_{\text{init}}, m, n, 0)$
Halting condition:	$M(\ell_{\text{halt}}, 0, 0, 0)$

Table 11.1: Encoding of the basic two-counter-machine instructions using difference constraints.

Lemma 11.1.3. *Suppose we are given a sentence $\varphi_{\mathcal{M}}$ encoding the behavior of a two-counter machine $\mathcal{M}$ as described above, then the sentence $M(\ell_{init}, m, n, 0) \wedge \varphi_{\mathcal{M}} \wedge \neg M(\ell_{halt}, 0, 0, 0)$ is unsatisfiable if and only if $\mathcal{M}$ halts when started on any given input $\langle m, n \rangle$.*

Notice that the sentence in Lemma 11.1.3 could also be transformed into a Horn sentence. Hence, by Proposition 11.1.2, it follows that satisfiability for Presburger arithmetic sentences restricted to difference constraints (requiring only the constant 1 besides the input) plus a single 4-ary uninterpreted predicate symbol is undecidable.

Encoding two-counter machines using quotient constraints works very similarly. We only need to change the representation of counter values in a state $M(\ell, x, y, z)$ as follows: $c_1 = \log_2(\frac{x}{2^z}) = \log_2(x) - \log_2(z) - 1$ and $c_2 = \log_2(\frac{y}{2^z}) = \log_2(y) - \log_2(z) - 1$. Incrementing the first counter is encoded by the sentence $\forall xyzx'.\ x' = 2 \cdot x \wedge M(\ell, x, y, z) \rightarrow M(\ell + 1, x', y, z)$, and the conditional decrement instruction for the first counter is encoded by

$$\big(\forall xyzy'z'.\, x > 2 \cdot z\ \wedge\ y' = 2 \cdot y\ \wedge\ z' = 2 \cdot z\ \wedge\ M(\ell, x, y, z)\ \rightarrow\ M(\ell + 1, x, y', z')\big)$$
$$\wedge\ \big(\forall xyz.\, x = 2 \cdot z\ \wedge\ M(\ell, x, y, z)\ \rightarrow\ M(\ell', x, y, z)\big)\ .$$

Analogous to the case of difference constraints, we thus infer undecidability of the satisfiability problem for Presburger arithmetic restricted to quotient constraints (requiring only the constant 2) and a single uninterpreted 4-ary predicate symbol. In this encoding incrementing a counter amounts to multiplying the corresponding component of M by two.

If we leave the realm of the integers and go to the rational numbers or the reals, we could represent an increment operation with division by two. This means that we actually do not have to leave the unit interval and still represent arbitrarily large counter values. More technically, the current value of counter C_1 would be $c_1 = -\log_2(\frac{2x}{z}) = -\log_2(x) + \log_2(z) - 1$. Then, incrementing C_1 is encoded by the sentence $\forall xyzx'.\, 2 \cdot x' = x \wedge M(\ell, x, y, z) \rightarrow M(\ell + 1, x', y, z)$, and $\texttt{test\&dec}(C_1, \ell')$ is encoded by

$$\begin{aligned}&\big(\forall xyzy'z'.\, 2 \cdot x < z \ \wedge\ 2 \cdot y' = y \ \wedge\ 2 \cdot z' = z \ \wedge\ M(\ell, x, y, z) \ \rightarrow\ M(\ell + 1, x, y', z')\big)\\ &\wedge \big(\forall xyz.\, 2 \cdot x = z \ \wedge\ M(\ell, x, y, z) \ \rightarrow\ M(\ell', x, y, z)\big)\ .\end{aligned}$$

In this encoding, we can limit the range of the last three components of M to the rational or real unit interval $[0, 1)$, i.e. they are bounded from below and above. Nevertheless, the associated satisfiability problem is undecidable.

Having additive constraints of the form $x + y \triangleleft c$ at hand, we can simulate subtraction by defining the additive inverse using an atom $x + x_- = 0$. To keep track of inverses, we adjust the arity of M accordingly and add the side condition

$$\forall xx_-yy_-zz_-.\, M(\ell, x, x_-, y, y_-, z, z_-) \ \rightarrow\ x + x_- = 0 \ \wedge\ y + y_- = 0 \ \wedge\ z + z_- = 0\ .$$

Counter values are represented in the same way as we have done for difference constraints. The increment instruction for the first counter is thus encoded by the sentence

$$\begin{aligned}\forall xx_-yy_-zz_-x'x'_-.\, x' + x_- = 1 \ \wedge\ x' + x'_- = 0 \ \wedge\ M(\ell, x, x_-, y, y_-, z, z_-)&\\ \rightarrow\ M(\ell + 1, x', x'_-, y, y_-, z, z_-)\ .&\end{aligned}$$

It is now straightforward to come up with the encoding of the conditional decrement. Hence, satisfiability for Presburger arithmetic restricted to additive constraints and a single free predicate symbol of arity 7 is undecidable. However, this time we need two constants, namely 1 and 0.

In order to complete the picture, we leave the realm of linear arithmetic for a little while and consider multiplicative constraints of the form $x \cdot y \triangleleft c$. These relate to quotient constraints like additive constraints relate to difference constraints. Hence, combining the previously used ideas of offsets and inverses, we can encode two-counter machines also with multiplicative constraints:

$$\begin{aligned}\forall xx_{-1}yy_{-1}zz_{-1}x'x'_{-1}.\, x \cdot x'_{-1} = 2 \ \wedge\ x' \cdot x'_{-1} = 1 \ \wedge\ M(\ell, x, x_{-1}, y, y_{-1}, z, z_{-1})&\\ \rightarrow\ M(\ell + 1, x', x'_{-1}, y, y_{-1}, z, z_{-1})&\end{aligned}$$

encodes the increment instruction on the first counter, for instance, using the rationals or reals as domain. As in the case of quotient constraints, we could restrict the range of variables to $(0, 1]$. Consequently, this yields another fragment of Rational arithmetic with uninterpreted predicate symbols for which the satisfiability problem is undecidable.

Theorem 11.1.4. *The satisfiability problem associated with Presburger arithmetic plus uninterpreted predicate symbols is undecidable, even if we restrict arithmetic atoms to difference constraints, additive constraints, or quotient constraints. Over the domain of rational or real numbers, we have the same undecidability results for the same fragments, plus the fragment with multiplicative constraints (which is nonlinear). In addition, in the case of quotient and multiplicative constraints, the result still holds if we restrict the domain to the rational or real unit interval.*

11.2 Encoding Two-Counter Machine Computations Using a Single Unary Predicate

It turns out that it is sufficient to add a single uninterpreted unary predicate symbol P to the vocabulary of Presburger arithmetic to facilitate encodings of two-counter machines, their

computations, and the associated halting problem. As soon as we have constructed a $\Sigma_{\mathrm{PA}+P}$-sentence φ that encodes the behavior of a given machine $\mathcal{M}$ together on a given input pair $\langle m, n\rangle$, we are interested in the (un)satisfiability of φ. Hence, we pose the question: Is there a $\Sigma_{\mathrm{PA}+P}$-structure $\mathcal{A}$ (extending the integers with addition, subtraction, and order relations) with $P^{\mathcal{A}} \subseteq \mathbb{Z}$ such that $\mathcal{A} \models \varphi$, or is there no such structure? For the sake of simplicity, we shall restrict the domain we consider to $\mathbb{N} = \mathbb{Z}_{\geq 0}$ most of the time, if not explicitly stated otherwise. We shall first give an informal description in the next section, and then we will get more formal in Section 11.2.2. Sections 11.2.3 and 11.2.4 are devoted to encoding variants that use a minimal number of quantifiers. In Section 11.2.5, we shall transfer our undecidability result to the rational domain, and in Section 11.2.6 we will discuss the case where P is replaced with an uninterpreted function symbol.

11.2.1 Informal Description of the Encoding

Since any interpretation $P^{\mathcal{A}}$ of the predicate symbol P is a subset of the natural numbers, we can conceive $P^{\mathcal{A}}$ as an infinite sequence of bits $b_0 b_1 b_2 \ldots$, where for every $n \in \mathbb{N}$ we have

$$b_n := \begin{cases} 0 & \text{if } n \notin P^{\mathcal{A}}, \\ 1 & \text{if } n \in P^{\mathcal{A}}. \end{cases}$$

Given a two-counter machine $\mathcal{M}$ with $K+1$ program lines, labeled $0, \ldots, K$, and two input values m, n, we shall represent in $P^{\mathcal{A}}$ all the configurations that occur during the run of $\mathcal{M}$ when started on input $\langle m, n\rangle$. One such configuration consists of the label of the program line that is to be executed in the next step, the current value c_1 of the first counter C_1, and the current value c_2 of the second counter C_2.

We divide the bit sequence $P^{\mathcal{A}}$ into chunks of growing length, each delimited by the bit sequence 001011. Such a chunk is divided into three subchunks, using the bit sequence 0011 as a delimiter. The first subchunk contains the current program line encoded in unary. The second and third subchunks store the current values of the counters C_1, C_2, respectively, also encoded in unary notation. Hence, every chunk has the form

$$\underbrace{001011}_{\substack{\text{left de-}\\ \text{limiter}}}\, 1^{\ell} 0 \ldots 0\, \underbrace{0011}_{\substack{\text{first sub-}\\ \text{delimiter}}}\, 1^{c_1} 0 \ldots 0\, \underbrace{0011}_{\substack{\text{second}\\ \text{subde-}\\ \text{limiter}}}\, 1^{c_2} 0 \ldots 0\ ,$$

where ℓ is the label of the program line to be executed next, c_1 is the value currently stored in counter C_1, and c_2 is the value currently stored in counter C_2. The subsequences 1^{ℓ}, 1^{c_1} and 1^{c_2} are followed by blocks of zeros that fill up the gap before the next 0011 delimiter (indicating the start of the subsequent subchunk) or the next 001011 delimiter (indicating the beginning of the successor configuration).

We devise the encoding in such a way that the length of each chunk and its subchunks increases with the number of computation steps that have already been performed. This makes sure that there is always enough space available to store the current counter values, which may thus become arbitrarily large. Of course, we have to provide sufficient space in the beginning such that the label of any program line and the initial counter values m and n may be stored. In order to achieve this, we define the constant $d := \max\{K, m, n\} + 6$ and require that the leftmost chunk starts at position d, i.e. there is a 001011 delimiter starting at position d but none starting left of d.[3] The first three subchunks have length d each. Thus, the second chunk starts at position $4d$. The subchunks of the second chunk, however, shall have a length of $4d$ each.[4] Hence, the total length

[3] Using d as a starting point instead of 0, say, is convenient, since we can use this information about the starting point to determine the length of subchunks.

[4] Technically, a length of $d+1$ for the subchunks of the second chunk would suffice. After all, the value of a counter can increase by at most one in a single computation step. However, we have chosen to increase the length in an exponential fashion rather than a linear one, as this will keep the encoding simple.

of the second chunk is $12d$. This scheme shall continue indefinitely, i.e. the starting points of the chunks in the bit sequence are $d, 4d, 16d, 64d, 256d$, and so on. Consequently, all the chunks are large enough to store all possibly occurring counter values, as these can increase by at most one in every step of the computation. Figure 11.1 illustrates the structure of a single chunk in the sequence, starting at position x.

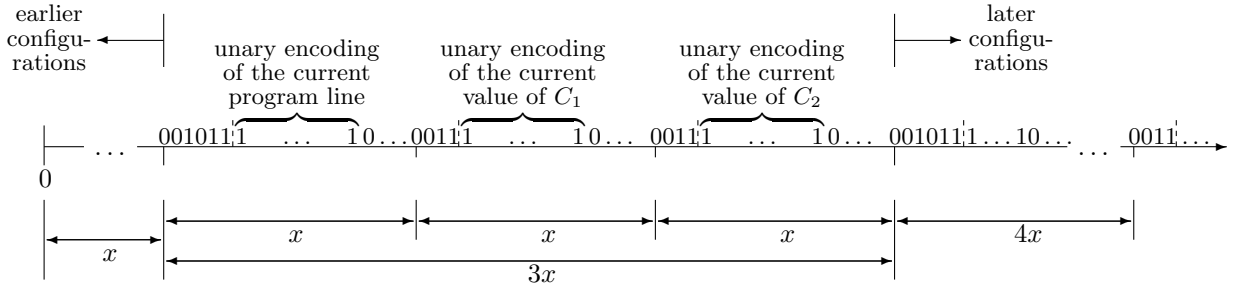


Figure 11.1: Structure of a single chunk of length $3x$.

11.2.2 Formal Encoding of Two-Counter Machine Computations

Recall that we assume to be given a two-counter machine $\mathcal{M}$ with $K+1$ program lines, labeled $0, \ldots, K$, and two input values m and n. We use the following abbreviations for arbitrary terms t:

$$\begin{aligned}
\psi_{001011}(t) &:= \neg P(t) \ \wedge\ \neg P(t+1) \ \wedge\ P(t+2) \ \wedge\ \neg P(t+3) \ \wedge\ P(t+4) \ \wedge\ P(t+5) \\
\psi_{0011}(t) &:= \neg P(t) \ \wedge\ \neg P(t+1) \ \wedge\ P(t+2) \ \wedge\ P(t+3) \\
\psi_{01}(t) &:= \neg P(t) \ \wedge\ P(t+1) \\
\psi_{10}(t) &:= P(t) \ \wedge\ \neg P(t+1) \\
\chi_\ell(t) &:= \psi_{10}(t+5+\ell) \qquad\qquad \text{for } \ell = 0, \ldots, K
\end{aligned}$$

First of all, we set up the general structure of the predicate P. Let d denote the integer with the value $d := \max\{K+6, m+4, n+4\}$. We use d as the starting point of our encoding.

$$\begin{aligned}
\varphi_1 :=\ & \psi_{001011}(d) && (11.1)\ \ \varphi_1 \\
& \wedge\ \big(\forall x.\ x < d \ \longrightarrow\ \neg P(x)\big) && (11.2) \\
& \wedge\ \big(\forall x.\ \psi_{001011}(x) \ \longrightarrow\ \psi_{0011}(2x) \ \wedge\ \psi_{0011}(3x) \ \wedge\ \psi_{001011}(4x)\big) && (11.3) \\
& \wedge\ \big(\forall xy.\ \psi_{001011}(x) \ \wedge\ \psi_{001011}(y) \ \wedge\ x \le y \ \wedge\ y < 4x \ \longrightarrow\ x = y\big) && (11.4) \\
& \wedge\ \big(\forall xy.\ \psi_{001011}(x) \ \wedge\ \psi_{0011}(y) \ \wedge\ x \le y \ \longrightarrow\ y \ge 2x\big) && (11.5) \\
& \wedge\ \big(\forall xy.\ \psi_{001011}(x) \ \wedge\ \psi_{0011}(y) \ \wedge\ 2x < y \ \longrightarrow\ y \ge 3x\big) && (11.6) \\
& \wedge\ \big(\forall xy.\ \psi_{001011}(x) \ \wedge\ \psi_{0011}(y) \ \wedge\ 3x < y \ \longrightarrow\ y \ge 4x\big) && (11.7) \\
& \wedge\ \big(\forall xy.\ \psi_{001011}(x) \ \wedge\ \psi_{01}(y) \ \wedge\ x+5 < y \ \wedge\ y < 4x \ \longrightarrow\ \psi_{0011}(y-1)\big) && (11.8)
\end{aligned}$$

Subformula (11.1) sets the first 001011 delimiter at position d and Subformula (11.2) ensures that this is indeed the leftmost such delimiter. Subormula (11.3) sets up all the other delimiters and Subformulas (11.4) to (11.7) guarantee that there are no spurious delimiters in between them. Subformula (11.8) stipulates that every 01 subsequence is part of one of the delimiters, i.e. there cannot be a subsequence 01 that lies outside of a 001011 or 0011 delimiter. This does also entail that between one delimiter (001011 or 0011) and the subsequent one there is exactly one subsequence 10, possibly overlapping with the last or first bit of one of the delimiters. Hence, this subsequence uniquely marks the end of the number encoded in the respective subchunk.

The following formula sets the initial values of the counters. Moreover, it sets the initial program line, which we assume to be zero:

$\varphi_2^{m,n}$

$$\varphi_2^{m,n} := \ \chi_0(d) \ \wedge \ \psi_{10}(2d+3+m) \ \wedge \ \psi_{10}(3d+3+n) \ .$$

Regarding the encoding of program lines, we have to enforce that the current program line never exceeds K. This is easily done with the formula

φ_3^K

$$\varphi_3^K := \ \forall xy.\, \psi_{001011}(x) \ \wedge \ \psi_{10}(y) \ \wedge \ x+5 \leq y \ \wedge \ y \leq 2x \ \longrightarrow \ y \leq x+5+K \ .$$

The previous formulas already ensure that exactly one label of a program line is encoded in every chunk.

Next we encode the control flow of $\mathcal{M}$. We assume that the following instructions occur in program line ℓ for some $\ell \in \{0, \ldots, K\}$.

Encoding of the instruction ℓ : `inc(`C_1`)`:

$$\begin{aligned}\forall xyz.\, \psi_{001011}(x) \ \wedge \ 2x \leq y \ \wedge \ y \leq 3x \ \wedge \ \psi_{10}(y) \ \wedge \ 3x \leq z \ \wedge \ z \leq 4x \ \wedge \ \psi_{10}(z) \ \wedge \ \chi_\ell(x) \\ \longrightarrow \ \psi_{10}(6x+y+1) \ \wedge \ \psi_{10}(9x+z) \ \wedge \ \chi_{\ell+1}(4x)\end{aligned}$$

The subfomula $\psi_{001011}(x)$ in the premise of the implication states that the chunk encoding the currently regarded configuration starts at position x. The other preconditions make clear that y and z correspond to the positions at which we find 10 subsequences in the two subchunks storing the current counter values:

$$\begin{array}{ccccccc} x & & 2x & & y & 3x & z \\ \downarrow & & \downarrow & & \downarrow & \downarrow & \downarrow \\ \underbrace{001011}_{\text{left delimiter}} 1^\ell 0 \ldots 0 & & \underbrace{0011}_{\text{first subdelimiter}} & 1^{c_1-1} 10 \ldots 0 & & \underbrace{0011}_{\text{second subdelimiter}} & 1^{c_2-1} 10 \ldots 0 \end{array}$$

Hence, C_1 and C_2 currently store the values $c_1 = y - 2x - 3$ and $c_2 = z - 3x - 3$, respectively. Since the subsequent chunk starts at position $4x$ and its second and third subchunks start at positions $8x$ and $12x$, respectively, we know that there must be one 10 subsequence at position $8x + 3 + c_1 + 1 = 6x + y + 1$ — the first counter is incremented by 1 — and one 10 subsequence must be at position $12x + 3 + c_2 = 9x + z$ — the value of the second counter remains unchanged. Moreover, the machine currently executes program line ℓ and is to continue at program line $\ell + 1$. Therefore, we put the formula $\chi_\ell(x)$ in the premise and the formula $\chi_{\ell+1}(4x)$ into the consequent of the implication.

Encoding of the instruction ℓ : `inc(`C_2`)`:

$$\begin{aligned}\forall xyz.\, \psi_{001011}(x) \ \wedge \ 2x \leq y \ \wedge \ y \leq 3x \ \wedge \ \psi_{10}(y) \ \wedge \ 3x \leq z \ \wedge \ z \leq 4x \ \wedge \ \psi_{10}(z) \ \wedge \ \chi_\ell(x) \\ \longrightarrow \ \psi_{10}(6x+y) \ \wedge \ \psi_{10}(9x+z+1) \ \wedge \ \chi_{\ell+1}(4x) \ .\end{aligned}$$

Encoding of the instruction ℓ : `test&dec(`C_1`,`ℓ'`)`:

The case of C_1 storing 0:

$$\begin{aligned}\forall xyz.\, \psi_{001011}(x) \ \wedge \ 2x \leq y \ \wedge \ y \leq 3x \ \wedge \ \psi_{10}(y) \ \wedge \ 3x \leq z \ \wedge \ z \leq 4x \ \wedge \ \psi_{10}(z) \ \wedge \ \chi_\ell(x) \\ \wedge \ y = 2x+3 \\ \longrightarrow \ \psi_{10}(6x+y) \ \wedge \ \psi_{10}(9x+z) \ \wedge \ \chi_{\ell'}(4x) \ .\end{aligned}$$

The condition $y = 2x + 3$ ensures that the first counter stores the value 0.

The case of C_1 storing a value greater than 0:

$$\forall xyz.\, \psi_{001011}(x) \wedge 2x \leq y \wedge y \leq 3x \wedge \psi_{10}(y) \wedge 3x \leq z \wedge z \leq 4x \wedge \psi_{10}(z) \wedge \chi_\ell(x) \\ \wedge\ y > 2x + 3 \\ \longrightarrow\ \psi_{10}(6x + y - 1) \wedge \psi_{10}(9x + z) \wedge \chi_{\ell+1}(4x)\ .$$

The condition $y > 2x + 3$ ensures that the first counter stores a value strictly greater than 0.

Encoding of the instruction ℓ : `test&dec(`C_2`,`ℓ'`)`:

The case of C_2 storing 0:

$$\forall xyz.\, \psi_{001011}(x) \wedge 2x \leq y \wedge y \leq 3x \wedge \psi_{10}(y) \wedge 3x \leq z \wedge z \leq 4x \wedge \psi_{10}(z) \wedge \chi_\ell(x) \\ \wedge\ z = 3x + 3 \\ \longrightarrow\ \psi_{10}(6x + y) \wedge \psi_{10}(9x + z) \wedge \chi_{\ell'}(4x)\ .$$

The case of C_2 storing a value greater than 0:

$$\forall xyz.\, \psi_{001011}(x) \wedge 2x \leq y \wedge y \leq 3x \wedge \psi_{10}(y) \wedge 3x \leq z \wedge z \leq 4x \wedge \psi_{10}(z) \wedge \chi_\ell(x) \\ \wedge\ z > 3x + 3 \\ \longrightarrow\ \psi_{10}(6x + y) \wedge \psi_{10}(9x + z - 1) \wedge \chi_{\ell+1}(4x)\ .$$

Encoding of the instruction ℓ : `goto(`ℓ'`)`:

$$\forall xyz.\, \psi_{001011}(x) \wedge 2x \leq y \wedge y \leq 3x \wedge \psi_{10}(y) \wedge 3x \leq z \wedge z \leq 4x \wedge \psi_{10}(z) \wedge \chi_\ell(x) \\ \longrightarrow\ \psi_{10}(6x + y) \wedge \psi_{10}(9x + z) \wedge \chi_{\ell'}(4x)\ .$$

Encoding of the instruction ℓ : `halt`:

$$\forall xyz.\, \psi_{001011}(x) \wedge 2x \leq y \wedge y \leq 3x \wedge \psi_{10}(y) \wedge 3x \leq z \wedge z \leq 4x \wedge \psi_{10}(z) \wedge \chi_\ell(x) \\ \longrightarrow\ \psi_{10}(6x + y) \wedge \psi_{10}(9x + z) \wedge \chi_K(4x)\ .$$

The consequent of the implication ensures that the counters remain unchanged and that the computation continues at program line K. Since we assume the K-th program line to contain the instruction `halt`, the rest of the bit sequence will repeat the same chunk structure again and again, as the counter values will remain unchanged and the encoded program line will also repeat indefinitely.

Finally, we pose the central question concerning the halting behavior of the machine: Does the machine ever reach a program line containing the `halt` instruction? The question is posed as a requirement in a negative fashion:

$$\varphi_4^K := \ \forall x.\, \psi_{001011}(x) \ \longrightarrow\ \neg\chi_K(x)\ . \qquad \varphi_4^K$$

Technically speaking, we require that the machine never reaches the K-th program line, which we assume to be the one and only line containing the `halt` instruction.

Given the the two-counter machine $\mathcal{M}$, we denote by $\varphi_\mathcal{M}$ the sentence that encodes $\mathcal{M}$'s behavior in accordance with the described formula schemes. Then, the sentence $\varphi_1 \wedge \varphi_2^{m,n} \wedge \varphi_3^K \wedge \varphi_4^K \wedge \varphi_\mathcal{M}$ is satisfied if and only if the machine will never reach the instruction `halt` when started on the given input. $\varphi_\mathcal{M}$

Lemma 11.2.1. *The two-counter machine $\mathcal{M}$ with $K + 1$ program lines, labeled $0, \ldots, K$, started on input $\langle m, n\rangle$ eventually reaches a program line containing the instruction* `halt` *if and only if the $\Sigma_{\mathrm{PA}+P}$-sentence $\varphi_1 \wedge \varphi_2^{m,n} \wedge \varphi_3^K \wedge \varphi_4^K \wedge \varphi_\mathcal{M}$ is unsatisfiable.*

Proof sketch. We first observe the following technical properties of every structure $\mathcal{A}$ with $\mathcal{A} \models \varphi_1$.

(a) For every integer $r \in \mathbb{N}$ we have $\mathcal{A} \models \psi_{001011}(r)$ if and only if $r = 4^i d$ for some $i \in \mathbb{N}$.

(b) For every integer $r \in \mathbb{N}$ we have $\mathcal{A} \models \psi_{0011}(r)$ if and only if $r = 2 \cdot 4^i d$ or $r = 3 \cdot 4^i d$ for some $i \in \mathbb{N}$.

(c) For every integer $r \in \mathbb{N}$ we have $\mathcal{A} \models \psi_{01}(r)$ if and only if

$$r \in \bigcup_{i \in \mathbb{N}} \left\{ 4^i d + 1,\ 4^i d + 3,\ 2 \cdot 4^i d + 1,\ 3 \cdot 4^i d + 1 \right\} .$$

(d) Suppose there are integers $i, r, q \in \mathbb{N}$ such that $4^i d + 5 \leq r, q < 2 \cdot 4^i d$. If we have $\mathcal{A} \models \psi_{10}(r)$ and $\mathcal{A} \models \psi_{10}(q)$, then it follows that $r = q$.

(e) Suppose there are integers $i, r, q \in \mathbb{N}$ such that $2 \cdot 4^i d + 3 \leq r, q < 3 \cdot 4^i d$. If we have $\mathcal{A} \models \psi_{10}(r)$ and $\mathcal{A} \models \psi_{10}(q)$, then it follows that $r = q$.

(f) Suppose there are integers $i, r, q \in \mathbb{N}$ such that $3 \cdot 4^i d + 3 \leq r, q < 4^{i+1} d$. If we have $\mathcal{A} \models \psi_{10}(r)$ and $\mathcal{A} \models \psi_{10}(q)$, then it follows that $r = q$.

(g) For every integer $i \in \mathbb{N}$ there are integers $r_1, r_2, r_3 \in \mathbb{N}$ such that

- $4^i d + 5 \leq r_1 < 2 \cdot 4^i d$ and $\mathcal{A} \models \psi_{10}(r_1)$,
- $2 \cdot 4^i d + 3 \leq r_2 < 3 \cdot 4^i d$ and $\mathcal{A} \models \psi_{10}(r_2)$, and
- $3 \cdot 4^i d + 3 \leq r_3 < 4^{i+1} d$ and $\mathcal{A} \models \psi_{10}(r_3)$.

Due to the above observations, it is clear that any model $\mathcal{A}$ of φ_1 interprets P in such a way that it uniquely represents an infinite sequence of triples of nonnegative integers encoded in unary, just as we have described it earlier (cf. Figure 11.1). If, in addition, $\mathcal{A}$ satisfies $\varphi_2^{m,n}$ and φ_3^K, then the first triple of the sequence has the form $\langle 0, m, n \rangle$ and the first component of every triple in the sequence does not exceed K. Then, for any model $\mathcal{A} \models \varphi_1 \wedge \varphi_2^{m,n} \wedge \varphi_3^K \wedge \varphi_{\mathcal{M}}$ the interpretation $P^{\mathcal{A}}$ of P does not only represent a sequence of triples of integers but also establishes relations between the triples in the sequence, such that they mimic $\mathcal{M}$'s behavior. The only technical difference is that whenever $\mathcal{M}$ enters a configuration $\langle \ell, c_1, c_2 \rangle$ such that program line ℓ contains `halt`, then all later configurations have the form $\langle K, c_1, c_2 \rangle$. All in all, $P^{\mathcal{A}}$ is a faithful encoding of some run of $\mathcal{M}$ starting from the input $\langle m, n \rangle$.

On the other hand, since $\mathcal{M}$ is deterministic, there is a unique sequence

$$\tau := \langle \ell_{\text{init}}, m, n \rangle \langle \ell_1, c_{1,1}, c_{2,1} \rangle \langle \ell_2, c_{1,2}, c_{2,2} \rangle \langle \ell_3, c_{1,3}, c_{2,3} \rangle \dots$$

of configurations that represents the *run of $\mathcal{M}$ started on input $\langle m, n \rangle$*. If τ is finite and thus contains a halting configuration $\langle \ell, c_1, c_2 \rangle$ as its last triple, we concatenate the infinite sequence $\langle K, c_1, c_2 \rangle \langle K, c_1, c_2 \rangle \dots$ and thus obtain an infinite sequence again. This infinite sequence (be it originally infinite or made so artificially) can be translated into a structure $\mathcal{A}_\tau$ such that $\mathcal{A}_\tau \models \varphi_1 \wedge \varphi_2^{m,n} \wedge \varphi_3^K \wedge \varphi_{\mathcal{M}}$.

So far, we have seen that $\varphi_1 \wedge \varphi_2^{m,n} \wedge \varphi_3^K \wedge \varphi_{\mathcal{M}}$ is satisfiable and that every model represents the unique run τ of $\mathcal{M}$ started on input $\langle m, n \rangle$. Then, we observe for any model $\mathcal{A} \models \varphi_1 \wedge \varphi_2^{m,n} \wedge \varphi_3^K \wedge \varphi_{\mathcal{M}}$ that $\mathcal{A} \models \varphi_4^K$ holds if and only if τ *does not* contain a triple $\langle K, c_1, c_2 \rangle$ for any $c_1, c_2 \in \mathbb{N}$. Hence, $\varphi_1 \wedge \varphi_2^{m,n} \wedge \varphi_3^K \wedge \varphi_{\mathcal{M}} \wedge \varphi_4^K$ is unsatisfiable if and only if $\mathcal{M}$ reaches the `halt` instruction when started on the input $\langle m, n \rangle$. □

Together with the fact that the halting problem for two-counter machines is undecidable (cf. Proposition 11.1.2), we get the following theorem.

Theorem 11.2.2. *(Un)satisfiability of the universal fragment of* PA+P *is undecidable.*

11.2.3 Reducing the Number of Variables to Two

We can formulate the encoding with at most two variables per subformula. All we have to do are little modifications of the encodings of the two-counter machine instructions.

Modified encoding of the instruction ℓ : `inc(`C_1`)`:

$$\forall xy.\ \psi_{001011}(x) \ \wedge\ 2x \leq y \ \wedge\ y \leq 3x \ \wedge\ \psi_{10}(y) \ \wedge\ \chi_\ell(x) \ \longrightarrow\ \psi_{10}(6x+y+1) \ \wedge\ \chi_{\ell+1}(4x)$$

$$\forall xz.\ \psi_{001011}(x) \ \wedge\ 3x \leq z \ \wedge\ z \leq 4x \ \wedge\ \psi_{10}(z) \ \wedge\ \chi_\ell(x) \ \longrightarrow\ \psi_{10}(9x+z)$$

For this instruction and most of the others we split the encoding formula into two parts: the first formula realizes the y-part of the original encoding and the second formula realizes the z-part.

Modified encoding of the instruction ℓ : `inc(`C_2`)`:

$$\forall xy.\ \psi_{001011}(x) \ \wedge\ 2x \leq y \ \wedge\ y \leq 3x \ \wedge\ \psi_{10}(y) \ \wedge\ \chi_\ell(x) \ \longrightarrow\ \psi_{10}(6x+y) \ \wedge\ \chi_{\ell+1}(4x)$$

$$\forall xz.\ \psi_{001011}(x) \ \wedge\ 3x \leq z \ \wedge\ z \leq 4x \ \wedge\ \psi_{10}(z) \ \wedge\ \chi_\ell(x) \ \longrightarrow\ \psi_{10}(9x+z+1)$$

Modified encoding of the instruction ℓ : `test&dec(`C_1`,`ℓ'`)`:

The case of C_1 storing 0:

$$\forall xz.\ \psi_{001011}(x) \ \wedge\ 3x \leq z \ \wedge\ z \leq 4x \ \wedge\ \psi_{10}(z) \ \wedge\ \chi_\ell(x) \ \wedge\ \psi_{10}(2x+3) \ \longrightarrow\ \psi_{10}(8x+3) \ \wedge\ \psi_{10}(9x+z) \ \wedge\ \chi_{\ell'}(4x)$$

The subformula $\psi_{10}(2x+3)$ in the premise ensures that the counter C_1 currently stores a 0 and the subformula $\psi_{10}(8x+3)$ requires that C_1 still stores 0 in the next step. Notice that we do not need a variable y to address the corresponding bit positions, since we can directly compute these positions from x.

The case of C_1 storing a value greater than 0:

$$\forall xy.\ \psi_{001011}(x) \ \wedge\ 2x \leq y \ \wedge\ y \leq 3x \ \wedge\ \psi_{10}(y) \ \wedge\ \chi_\ell(x) \ \wedge\ y > 2x+3 \ \longrightarrow\ \psi_{10}(6x+y-1) \ \wedge\ \chi_{\ell+1}(4x)$$

$$\forall xz.\ \psi_{001011}(x) \ \wedge\ 3x \leq z \ \wedge\ z \leq 4x \ \wedge\ \psi_{10}(z) \ \wedge\ \chi_\ell(x) \ \wedge\ \neg\psi_{10}(2x+3) \ \longrightarrow\ \psi_{10}(9x+z)$$

In the first sentence $y > 2x+3$ ensures that the value of C_1 is greater than zero. In the second sentence C_1's exact value is not important and thus $\neg\psi_{10}(2x+3)$ is sufficient for ensuring that C_1's value is strictly positive.

Modified encoding of the instruction ℓ : `test&dec(`C_2`,`ℓ'`)`:

The case of C_2 storing 0:

$$\forall xy.\ \psi_{001011}(x) \ \wedge\ 2x \leq y \ \wedge\ y \leq 3x \ \wedge\ \psi_{10}(y) \ \wedge\ \chi_\ell(x) \ \wedge\ \psi_{10}(3x+3) \ \longrightarrow\ \psi_{10}(6x+y) \ \wedge\ \psi_{10}(12x+3) \ \wedge\ \chi_{\ell'}(4x)$$

The case of C_2 storing a value greater than 0:

$$\forall xy.\ \psi_{001011}(x) \ \wedge\ 2x \leq y \ \wedge\ y \leq 3x \ \wedge\ \psi_{10}(y) \ \wedge\ \chi_\ell(x) \ \wedge\ \neg\psi_{10}(3x+3) \ \longrightarrow\ \psi_{10}(6x+y) \ \wedge\ \chi_{\ell+1}(4x)$$

$$\forall xz.\ \psi_{001011}(x) \ \wedge\ 3x \leq z \ \wedge\ z \leq 4x \ \wedge\ \psi_{10}(z) \ \wedge\ \chi_\ell(x) \ \wedge\ z > 3x+3 \ \longrightarrow\ \psi_{10}(9x+z-1)$$

Modified encoding of the instruction $\ell :$ `goto`(ℓ'):

$$\forall xy.\ \psi_{001011}(x) \ \wedge\ 2x \leq y \ \wedge\ y \leq 3x \ \wedge\ \psi_{10}(y) \ \wedge\ \chi_\ell(x) \ \longrightarrow\ \psi_{10}(6x+y) \ \wedge\ \chi_{\ell'}(4x)$$

$$\forall xz.\ \psi_{001011}(x) \ \wedge\ 3x \leq z \ \wedge\ z \leq 4x \ \wedge\ \psi_{10}(z) \ \wedge\ \chi_\ell(x) \ \longrightarrow\ \psi_{10}(9x+z)$$

Modified encoding of the instruction $\ell :$ `halt`:

$$\forall xy.\ \psi_{001011}(x) \ \wedge\ 2x \leq y \ \wedge\ y \leq 3x \ \wedge\ \psi_{10}(y) \ \wedge\ \chi_\ell(x) \ \longrightarrow\ \psi_{10}(6x+y) \ \wedge\ \chi_K(4x)$$

$$\forall xz.\ \psi_{001011}(x) \ \wedge\ 3x \leq z \ \wedge\ z \leq 4x \ \wedge\ \psi_{10}(z) \ \wedge\ \chi_\ell(x) \ \longrightarrow\ \psi_{10}(9x+z)$$

Theorem 11.2.3. *(Un)satisfiability of the universal fragment of* PA+P *is undecidable, if we allow at least two quantifiers.*

11.2.4 Undecidability with One Variable Only Using Another Encoding

It is not obvious how to refine the encoding from the previous section in such a way that a single variable would suffice. However, the result of Theorem 11.2.3 can be improved, if we use a different approach. The following is an adaptation of Downey's encoding [Dow72]. It results in a sentence that could easily be transformed into one containing only a single quantifier. Let $p_1, \ldots, p_{K+4}$ be the $K+4$ smallest prime numbers. In the following sentences, we shall address them by $q_t, q_{C_1}, q_{C_2}, q_0, q_1, \ldots, q_K$ and use them as prime factors. A configuration of the encoded two-counter machine is described by a single natural number with the prime factorization $(q_t)^t \cdot q_\ell \cdot (q_{C_1})^{c_1} \cdot (q_{C_2})^{c_2}$, where t denotes the current time stamp, ℓ is the current program line, which is to be executed next, and c_1, c_2 are the current values of the two counters C_1, C_2, respectively. The sentences entail that P contains every reachable configuration.

Initial condition: $P\big(q_t \cdot q_{\ell_{\text{init}}} \cdot q_{C_1}^m \cdot q_{C_2}^n\big)$.

The sentence stipulates that the initial configuration $\langle 1, \ell_{\text{init}}, n, m\rangle$ can be reached.

Encoding of the instruction $\ell :$ `inc`(C_1): $\forall x.\, P(q_\ell \cdot x) \to P(q_t \cdot q_{\ell+1} \cdot q_{C_1} \cdot x)$.

If the machine can reach any configuration $\langle t, \ell, c_1, c_2\rangle$, then the machine can also reach the configuration $\langle t+1, \ell+1, c_1+1, c_2\rangle$.

Encoding of the instruction $\ell :$ `inc`(C_2): $\forall x.\, P(q_\ell \cdot x) \to P(q_t \cdot q_{\ell+1} \cdot q_{C_2} \cdot x)$.

Encoding of the instruction $\ell :$ `test&dec`(C_1, ℓ'):

$$\forall x.\ \bigwedge_{i=1}^{q_{C_1}-1} \Big(P\big(q_\ell \cdot (q_{C_1} \cdot x + i)\big) \to P\big(q_t \cdot q_{\ell'} \cdot (q_{C_1} \cdot x + i)\big)\Big)$$
$$\wedge\, \forall x.\, P(q_\ell \cdot q_{C_1} \cdot x) \to P(q_t \cdot q_{\ell+1} \cdot x)\ .$$

The first part of the sentence stipulates that, if the machine can reach any configuration $\langle t, \ell, 0, c_2\rangle$ where the counter C_1 is zero, then it can also reach $\langle t+1, \ell', 0, c_2\rangle$. Notice that any natural number $q_{C_1} \cdot x + i$ with $1 \leq i \leq q_{C_1} - 1$ is not divisible by q_{C_1}. Conversely, for every natural number $y > 0$ that is not divisible by q_{C_1} there are natural numbers $x \geq 0$ and $i \geq 1$ such that $y = q_{C_1} \cdot x + i$ and $1 \leq i \leq q_{C_1} - 1$. The second part says that, if the machine can reach any configuration $\langle t, \ell, c_1, c_2\rangle$ with $c_1 > 0$, then it can also reach the configuration $\langle t+1, \ell+1, c_1-1, c_2\rangle$.

Encoding of the instruction $\ell :$ `test&dec`(C_2, ℓ'):

$$\forall x.\ \bigwedge_{i=1}^{q_{C_2}-1} \Big(P\big(q_\ell \cdot (q_{C_2} \cdot x + i)\big) \to P\big(q_t \cdot q_{\ell'} \cdot (q_{C_2} \cdot x + i)\big)\Big)$$
$$\wedge\, \forall x.\, P(q_\ell \cdot q_{C_2} \cdot x) \to P(q_t \cdot q_{\ell+1} \cdot x)\ .$$

Encoding of the instruction ℓ : `goto`(ℓ'): $\forall x.\, P(q_\ell \cdot x) \to P(q_t \cdot q_{\ell'} \cdot x)$.

If the machine can reach any configuration $\langle t, \ell, c_1, c_2\rangle$, then it can also reach $\langle t+1, \ell', c_1, c_2\rangle$.

Encoding of the instruction ℓ : `halt`: $\forall x.\, P(q_\ell \cdot x) \to P(q_t \cdot q_\ell \cdot x)$.

If the machine can reach any configuration $\langle t, \ell, c_1, c_2\rangle$, then the machine will loop forever while staying at program line ℓ, which contains the instruction `halt`.

For any two-counter machine $\mathcal{M}$ with a program containing $K+1 \geq 2$ lines the computation of $\mathcal{M}$ on the input m, n can be formalized using the above encoding. Then, for any model $\mathcal{A}$ of the encoding, we have $\mathcal{A} \models P\big(q_t^t \cdot q_\ell \cdot q_{C_1}^{c_1} \cdot q_{C_2}^{c_2}\big)$ if and only if $\mathcal{M}$ reaches the configuration $\langle t, \ell, c_1, c_2\rangle$ at the t-th step of its run. Let $\varphi_{\mathcal{M},m,n}$ be the sentence resulting from the encoding. Then, we observe that the sentence $\varphi_{\mathcal{M},m,n} \wedge \forall x.\, \neg P(q_{\ell_{\text{halt}}} \cdot x)$ is satisfiable if and only if $\mathcal{M}$ does not halt when started on the input m, n. Notice that this sentence can be converted into an equivalent PA+P sentence in conjunctive normal form that is Horn and Krom and contains exactly one universal quantifier.

Theorem 11.2.4. *(Un)satisfiability of the universal fragment of* PA+P *is undecidable, even if we restrict the language to sentences that are Horn and Krom and contain only a single first-order variable.*

We have not yet explained why we have introduced the time stamps to the configurations. This allows keeping track of the sequence of configurations. In the context of the halting problem, this is not utterly important, as one is merely concerned with the reachability of the program line containing `halt`. However, in the context of the *recurrence problem*, this ability is crucial. The sentence $\varphi_{\mathcal{M},0,0} \wedge \forall x \exists y.\, x < y \wedge P(q_{\ell_{\text{init}}} \cdot y)$ is satisfiable if and only if $\mathcal{M}$'s run is recurring when started on the input $m = 0, n = 0$, i.e. if it reaches the program line ℓ_{init} infinitely often.

Proposition 11.2.5. *The* recurrence problem *for deterministic two-counter machines can be expressed in the Horn-Krom fragment of* PA+P *using a single* $\forall\exists$ *quantifier alternation and at most two variables per clause.*

The recurrence problem will be of importance in Section 11.3, where we shall use it to show that the satisfiability problem for PA+P with a single quantifier alternation is Σ_1^1-hard.

11.2.5 Using the Rationals or Reals as Underlying Domain

Presburger arithmetic is defined on the integers and we have shown that adding a single uninterpreted unary predicate symbol yields an undecidable satisfiability problem. We can directly use the encoding that we have presented for the integers in order to show undecidability over the rational and real domains. The crucial point is that we have encoded the reachability of the `halt` instruction in a negative fashion. If the machine $\mathcal{M}$ reaches a `halt` instruction, then we cannot find a model of the encoding sentence $\varphi_1 \wedge \varphi_2^{m,n} \wedge \varphi_3^K \wedge \varphi_\mathcal{M} \wedge \varphi_4^K$, since any structure that faithfully represents the run of $\mathcal{M}$ on the given input must violate the condition $\neg\chi_K(j)$ for some integer j for which $\psi_{001011}(j)$ is true. We have used this observation to prove Lemma 11.2.1. The described conflict does not vanish when we assume a larger domain. If, on the other hand, the machine $\mathcal{M}$ does not reach a `halt` instruction, then there is a model of $\varphi_1 \wedge \varphi_2^{m,n} \wedge \varphi_3^K \wedge \varphi_\mathcal{M} \wedge \varphi_4^K$. In particular, there is a model in which P is interpreted such that it exclusively contains integers and no reals at all. Hence, the fact that we are dealing with an extended domain does not affect the circumstances under which the encoding sentence is unsatisfiable or not. Consequently, we have the following undecidability result.

Theorem 11.2.6. *(Un)satisfiability of the universal fragment of linear arithmetic over the rationals or reals with a single uninterpreted unary predicate symbol is undecidable.*

11.2.6 Unary Function Symbols and the Horn Fragment

The uninterpreted unary predicate symbol P in our encoding of two-counter machines can be replaced with an uninterpreted unary function symbol $f : \mathbb{N} \to \mathbb{N}$ over the natural numbers. We simply add the assertion $\forall x.\, f(x) \leq 1$ and substitute every negative literal $\neg P(t)$ with $f(t) = 0$ and every positive literal $P(t)$ with $f(t) = 1$, where t is any term. (Implicitly, we exploit the fact that f is interpreted by a total function $f^{\mathcal{A}}$ in any structure $\mathcal{A}$.) After this substitution, transforming the encoding formula set from Section 11.2.2 into conjunctive normal form (CNF) yields a sentence that can easily be transformed into a *Horn* sentence, i.e. every clause contains at most one positive literal. The reason is that we can use negation very liberally: $\neg P(t)$ corresponds to $f(t) = 0$ and, at the same time, to $\neg f(t) = 1$; $P(t)$ corresponds to $f(t) = 1$ and, at the same time, to $\neg f(t) = 0$. By this line of argument we obtain the following theorem.

Theorem 11.2.7. *(Un)satisfiability of the universal Horn fragment of Presburger arithmetic with a single uninterpreted unary function symbol is undecidable.*

Over the domain of the reals, we can replace the predicate symbol P in the same spirit, yet in a slightly different way. For one thing, we add the assertion $\forall x.\, 0 \leq f(x) \;\wedge\; f(x) \leq 1$ to the encoding, which also introduces an explicit lower bound to the values of f. As this assertion alone does not guarantee that in any model the image of $f : \mathbb{R} \to \mathbb{R}$ contains at most two values, we replace any occurrence of $\neg P(t)$ with $f(t) = 0$ and any occurrence of $P(t)$ with $f(t) > 0$. Again, a CNF transformation yields a sentence that can be transformed into a Horn sentence.

Theorem 11.2.8. *(Un)satisfiability of the universal Horn fragment of linear arithmetic over the rationals or reals with a single additional uninterpreted unary function symbol is undecidable.*

11.3 Degrees of Unsolvability

We have shown that the unsatisfiability problem of the universal fragment of PA+P is undecidable. Next, we shall argue that the set of unsatisfiable sentences from this fragment is recursively enumerable. In order to prove this, it suffices to give a sound calculus that, given an unsatisfiable sentence over the language in question, derives `false` or the *empty clause* in finitely many steps. This property is known as *refutational completeness*. In fact, such a calculus would constitute a semi-decision procedure for unsatisfiable sentences.

refutational completeness

Indeed, *hierarchic superposition* [BGW94, BW13b, BW13a] (cf. Section 10.3) is such a refutationally complete calculus for all unsatisfiable *hierarchic clause sets* that are *sufficiently complete*, if the considered background theory is *compact* (cf. Theorem 24 in [BGW94]). The universal fragment of Presburger arithmetic with uninterpreted predicate symbols can be treated in this framework: We consider finite clause sets over the vocabulary Σ_{PA} enriched with arbitrary uninterpreted predicate symbols. All occurring first-order variables are implicitly universally quantified and we do not consider any uninterpreted constant or function symbols. Presburger arithmetic is conceived as the background theory and, hence, determines the interpretation of all symbols from Σ_{PA}. In this setting, the two requirements — sufficient completeness and compactness of the background theory — are satisfied. Sufficient completeness (cf. Definition 20 in [BGW94]) concerns uninterpreted constant and function symbols that range over the background sort. Since we do not allow such symbols in our language, all sentences are sufficiently complete. For the same reason, the background theory is compact. This means, every set of first-order sentences over Σ_{PA} that is not satisfied under $\mathbb{Z}$ has some finite (even a singleton) subset that is not satisfied under $\mathbb{Z}$. Hence, the following proposition holds.

Proposition 11.3.1. *The set of unsatisfiable sentences over the universal fragment of Presburger arithmetic with additional uninterpreted predicate symbols is recursively enumerable.*

From the literature on the *arithmetical hierarchy* (see, e.g. [Rog87, Soa87, Odi92, Soa16]) we get the following.[5]

[5] The sets Σ^0_n, Π^0_n are sets of sets of natural numbers that are describable by certain first-order-arithmetic

	Satisfiability	Unsatisfiability	Validity	Invalidity
$\forall^*$-fragment	Π^0_1-complete	Σ^0_1-complete	Σ^0_0	Σ^0_0
$\exists^*$-fragment	Σ^0_0	Σ^0_0	Σ^0_1-complete	Π^0_1-complete

Table 11.2: Overview regarding the degree of unsolvability of the (un)satisfiability and (in)validity problems for the purely universal and purely existential fragment of Presburger arithmetic with additional uninterpreted predicate symbols. Notice that membership in Σ^0_0 (which coincides with Π^0_0) entails decidability of the respective problem.

Proposition 11.3.2.

(i) The set Σ^0_1 captures exactly the recursively enumerable sets. Σ^0_1

(ii) The set Π^0_1 captures exactly the sets whose complement is recursively enumerable. Π^0_1

(iii) The halting problem for (ordinary) Turing machines is Σ^0_1-complete.

Proof. (i) and (ii) are reformulations of Theorems II.1.2 and IV.1.3 in [Soa87], respectively. (iii) combines the following parts of [Soa87]: Definitions I.3.1, I.4.1, I.4.5, Theorem II.4.2 and the discussion after Definition IV.2.1 on page 64. □

Since we have completed a chain of reductions from the halting problem of Turing machines via the halting problem of two-counter machines to the unsatisfiability problem of the universal fragment of Presburger arithmetic with uninterpreted predicate symbols, we conclude Σ^0_1-completeness of the latter problem by Lemma 11.2.1 together with Propositions 11.3.1 and 11.3.2.

Theorem 11.3.3. *The set of unsatisfiable sentences from the universal fragment of Presburger arithmetic with uninterpreted predicate symbols is Σ^0_1-complete.*

It is worth noticing that the theorem can be translated to the realm of linear arithmetic over the reals. The reason is that hierarchic superposition is also refutationally complete over the universal fragment of this language, if there are no uninterpreted constant or function symbols involved.

Since any reduction of a problem S to a problem T (both read as a set of Gödel numbers) at the same time yields a reduction from $\overline{S}$ to $\overline{T}$, the complement of a Σ^0_1-complete set is complete for Π^0_1. Hence, Theorem 11.3.3 entails Π^0_1-completeness of the set of satisfiable sentences over the same language.

There are strong ties between (un)satisfiability in the universal fragment of the language we consider and (in)validity in the dual language, the existential fragment. The bottom line is that the obtained completeness results can be transfered to the corresponding (in)validity problems. The overall situation is depicted in Table 11.2.

For the sake of completeness, we briefly discuss (un)satisfiability for the existential fragment. Kruglov and Weidenbach [KW12, Kru13] have presented a general result regarding the satisfiability problem for hierarchic clause sets that are ground. More precisely, they have devised a decision procedure for that problem, based on a hierarchic superposition calculus.

Proposition 11.3.4 (Corollary of Theorem 23 from [KW12])**.** *Satisfiability of the existential fragment of Presburger arithmetic with additional uninterpreted predicate symbols is decidable.*

With this knowledge we can complete the overview in Table 11.2 and thus reveal the full picture of where the (un)satisfiability and (in)validity problems of the universal and existential fragments of Presburger arithmetic augmented with uninterpreted predicate symbols reside in the arithmetical hierarchy.

formulas. Whenever we speak of problems or sets of sentences belonging to Σ^0_n or Π^0_n, or being complete for these classes, we implicitly refer to the Gödelization of these problems or sets of sentences.

One $\forall\exists$ Quantifier Alternation Yields Σ_1^1-Completeness

Halpern has shown that the satisfiability problem for Presburger arithmetic with any choice of additional uninterpreted function symbols and predicate symbols lies in Σ_1^1 in the *analytical hierarchy*[6] (Theorem 3.1 in [Hal91]). This result is independent of the number of occurring quantifier alternations. In the present section, we show that already a single quantifier alternation suffices to make the problem complete for Σ_1^1. We leverage the following result, due to Alur and Henzinger.

Proposition 11.3.5 (Lemma 8 in [AH94])**.** *The problem of deciding whether a given nondeterministic two-counter machine has a recurring computation is Σ_1^1-hard.*

A *nondeterministic two-counter machine* differs from the deterministic model described in Section 11.1 in that it allows nondeterministic branching after a program line has been executed. This means that after the execution of a program line ℓ (which does not result in a jump induced by a `test&dec` instruction) the machine does not necessarily proceed to the $(\ell+1)$-st line, but may have the choice between two specified options.

This kind of nondeterminism can easily be incorporated into the encoding presented in Section 11.2.2. For instance, the nondeterministic version of the instruction ℓ : `inc`(C_1) can be represented by the formula

$$\begin{aligned}\forall xyz.\, \psi_{001011}(x) \;\wedge\; 2x \leq y \;\wedge\; y \leq 3x \;\wedge\; \psi_{10}(y) \;\wedge\; 3x \leq z \;\wedge\; z \leq 4x \;\wedge\; \psi_{10}(z) \;\wedge\; \chi_\ell(x)\\ \longrightarrow\; \psi_{10}(6x+y+1) \;\wedge\; \psi_{10}(9x+z) \;\wedge\; \big(\chi_{\ell'}(4x) \vee \chi_{\ell''}(4x)\big)\;.\end{aligned}$$

The last conjunct $\big(\chi_{\ell'}(4x) \vee \chi_{\ell''}(4x)\big)$ now offers a choice between program lines ℓ' and ℓ'' as the ones that are to be executed next.

Consequently, we can reuse major parts of our encoding in order to prove Σ_1^1-hardness. For any nondeterministic two-counter machine $\mathcal{M}$ we write $\varphi'_{\mathcal{M}}$ to address the encoding of $\mathcal{M}$'s program in accordance with Section 11.2.2 and the just described adaptations due to the nondeterministic setting.

A run of a nondeterministic two-counter machine is considered to be *recurring* if and only if it starts with both counters set to zero and reaches the initial program line (with label 0) infinitely often. This means, we have to remove φ_4 from the encoding set of sentences and replace it with a proper formalization of the recurrence condition:

$$\varphi'_5 :=\; \forall x \exists y.\, x \leq y \;\wedge\; \psi_{001011}(y) \;\wedge\; \chi_0(y)\;.$$

This sentence formulates recurrence in a positive fashion by saying that at any point in time program line 0 will be reached eventually. Finally, in order to account for the specific input requirements posed in the definition of recurrence, we construct $\varphi_2^{0,0}$ rather than $\varphi_2^{m,n}$, i.e. we set $m = n = 0$.

Lemma 11.3.6. *The nondeterministic two-counter machine $\mathcal{M}$ has a recurring run if and only if $\varphi_1 \wedge \varphi_2^{0,0} \wedge \varphi_3^K \wedge \varphi'_{\mathcal{M}} \wedge \varphi'_5$ is satisfiable.*

By Proposition 11.3.5, this yields Σ_1^1-hardness. Due to the result by Halpern [Hal91], we know that the set of satisfiable Presburger arithmetic sentences with additional uninterpreted predicate symbols lies in Σ_1^1. Hence, we get the following theorem.

Theorem 11.3.7. *The set of satisfiable sentences of the $(\forall^*\exists)$-fragment of* PA$+P$ *is Σ_1^1-complete and, hence, neither it nor its complement are recursively enumerable.*

Notice that the theorem can be reformulated in terms of uninterpreted unary function symbols instead of uninterpreted unary predicate symbols. However, in contrast to Theorem 11.2.7, we lose

[6] See, e.g., [Rog87, Odi92] for a definition of the analytical hierarchy. It can be conceived as the second-order equivalent of the arithmetical hierarchy. The main result we need in the present thesis is that any problem that is hard or complete for Σ_n^1 or Π_n^1 with $n \geq 1$ is not recursively enumerable; the same applies to the complement of such a problem.

the property that the encoding results in a Horn sentence when transformed into CNF. The reason is the involved nondeterminism and the way we have encoded nondeterministic branching.

Over the domains of the rationals and reals, we can only show Σ_1^1-hardness of the satisfiability problem, since Halpern's upper bound only covers the realm of the natural numbers.

Theorem 11.3.8. *The set of satisfiable sentences of the* $(\forall^*\exists)$*-fragment of linear arithmetic over the rationals or reals with a single additional uninterpreted unary predicate symbol is* Σ_1^1*-hard and, hence, neither it nor its complement are recursively enumerable..*

Moreover, the encoding from Section 11.2.4 can be used to improve Theorem 11.3.7 and show Σ_1^1-completeness for the (non-Horn) $\forall\exists$-fragment of PA+P, cf. Proposition 11.2.5.

Theorem 11.3.9. *The set of satisfiable sentences of the* $\forall\exists$*-fragment of* PA+P *is* Σ_1^1*-complete and, hence, neither it nor its complement are recursively enumerable.*

11.4 An Encoding Based on Difference Constraints

We have seen in Section 11.1 that difference constraints together with uninterpreted predicate symbols yield an undecidable satisfiability problem. This contrasts our positive result from Section 10.4 concerning the satisfiability problem for finite BSR(BD) clause sets. To sharpen the contrast even further, we intend to show the following result in the present section. Even when we only consider the rational unit interval $[0, 1]$ as domain, adding arithmetic atoms of the form $x - y \lhd c$ to BSR(BD) where c is an uninterpreted constant symbol of sort $\mathbb{Q}$, yields an undecidable satisfiability problem. We shall show this in two ways, first via a simple encoding similar to the ones presented in Section 11.1, and then via an adaptation of the encoding from Section 11.2.3 in the language LRA+PN.

We start with a refinement of the simple encoding from Section 11.1. This time, we use a 5-ary uninterpreted predicate symbol $M : \mathbb{Q} \times \mathbb{Q} \times \mathbb{Q} \times \mathbb{Q} \times \mathbb{Q}$ to address the state of the two-counter machine as follows: $M(u, t, x, y, z)$ stands for a machine at instruction u and time step t where x and y store counter values relative to the offset z. We use an uninterpreted constant symbol c to determine the distance between two neighboring counter values.[7] For instance, an increment of counter C_1 amounts to adding c to the x-component. Together with the offset construction that we have already used in Section 11.1, this means that the counter values can be reconstructed from x, y, z as follows: $c_1 = \frac{1}{c}(x - z - c)$ and $c_2 = \frac{1}{c}(y - z - c)$.

We use the time stamp t to make sure that the value of c is chosen sufficiently small so as to keep all rationals occurring in the encoding of the run of $\mathcal{M}$ between 0 and 1. We stipulate that the `halt` instruction has to be encountered before time stamp 1 is reached. After reaching `halt` for the first time, all successive configurations will have the shape $M(\ell_{\text{halt}}, t, 0, 0, 0)$ where the time stamp t will keep on increasing by c until $t = 1$ is reached. Suppose $\mathcal{M}$ halts when started on the input m, n and further suppose that at most L increment operations are applied to any of the two counters. Then, in any model $\mathcal{A}$ of the corresponding encoding, $c^{\mathcal{A}}$ has to be such that $c^{\mathcal{A}} > 0$ and $c^{\mathcal{A}} \cdot \big(\max(m, n) + L + 1\big) < 1$. Of course, we also have to make sure that the labels for program lines stem from $[0, 1]$.

In Table 11.3 we give prototypical encodings of the instructions concerning counter C_1; the encoding for counter C_2 can be done analogously. The side conditions stipulated in φ_{side} make sure that (a) c is positive but less than 1, (b) for every time step t there is at most one configuration $\langle u, t, x, y, z\rangle$, and (c) the time difference between any two configurations is at least c. This ensures that there are no spurious configurations and that the run is finite.

Notice that the overall encoding sentence can be transformed into Horn form.

Remark 11.4.1. *A semi-decision procedure for finite satisfiable clause sets over the language of BSR(BD) plus the arithmetic atoms* $x - y \lhd c$ *with uninterpreted constant symbol* c *of sort* $\mathbb{Q}$ *that*

[7] The idea to use an uninterpreted Skolem constant as the quantity for increment was suggested to the author of the present thesis by Dietrich Kuske during breakfast at the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'17) in Reykjavik, Iceland, in June 2017.

Operation	Encoding
$\ell : \texttt{inc}(C_1)$	$\forall txyzt'x'.\, x' - x = c \wedge t' - t = c \wedge t' \leq 1 \wedge M(\ell, t, x, y, z) \rightarrow M(\ell + 1, t', x', y, z)$
$\ell : \texttt{test\&dec}(C_1, \ell')$	$\big(\forall txyzt'x'.\, x - z > c \wedge y' - y = c \wedge z' - z = c \wedge t' - t = c \wedge t' \leq 1 \wedge M(\ell, t, x, y, z)\big) \rightarrow M(\ell + 1, t', x, y', z')\big)$
	$\wedge \big(\forall txyzt'.\, x - z = c \wedge t' - t = c \wedge t' \leq 1 \wedge M(\ell, t, x, y, z) \rightarrow M(\ell', t', x, y, z)\big)$
$\ell : \texttt{goto}(\ell')$	$\forall txyzt'x'.\, t' - t = c \wedge t' \leq 1 \wedge M(\ell, t, x, y, z) \rightarrow M(\ell', t', x, y, z)$
$\ell_{\text{halt}} : \texttt{halt}$	$\forall txyzt'.\, t' - t = c \wedge M(\ell_{\text{halt}}, t, x, y, z) \rightarrow M(\ell_{\text{halt}}, t', 0, 0, 0)$
Initial condition:	$M(\ell_{\text{init}}, 0, (m+1) \cdot c, (n+1) \cdot c, 0)$
Halting condition:	$M(\ell_{\text{halt}}, 1, 0, 0, 0)$

Side conditions:

$$
\begin{aligned}
&\varphi_{\text{side}} := \\
&\quad 0 < c \wedge c < 1 \\
&\quad \wedge\ \forall utxyzu'x'y'z'.\, M(u,t,x,y,z) \wedge M(u',t,x',y',z') \rightarrow u = u' \wedge x = x' \wedge y = y' \wedge z = z' \\
&\quad \wedge\ \forall utxyzu't'x'y'z'.\, t \leq t' \wedge t' - t < c \wedge M(u,t,x,y,z) \wedge M(u',t',x',y',z') \rightarrow t = t' \\
&\quad \wedge\ \forall utxyz.\, M(u,t,x,y,z) \rightarrow t \leq 1 \wedge x \leq 1 \wedge y \leq 1 \wedge z \leq 1
\end{aligned}
$$

Table 11.3: Encoding of the basic two-counter-machine instructions, including a step counter.

is in addition conjoined with bounds $c_x \leq x \leq d_x$ and $c_y \leq y \leq d_y$ could proceed as follows. Let $\bar{\mathsf{c}}$ be the tuple collecting all occurring uninterpreted constant symbols. We enumerate all tuples $\bar{r}$ of rational numbers and use them as candidate values for the constant symbols in $\bar{\mathsf{c}}$. For each step of the enumeration, we consider the values of $\bar{\mathsf{c}} := \bar{r}$ to be fixed. After replacing the constant symbols from $\bar{\mathsf{c}}$ with the values from $\bar{r}$, we obtain a finite clause set that almost belongs to BSR(BD). We now compute the least common multiple of all denominators that occur in the clause set and multiply all rational constants with this value. After reduction of all rationals to integers we obtain an equisatisfiable finite clause set that belongs to BSR(BD). At this point, we apply the decision procedure from Section 10.4. If it succeeds, the clause set at hand is satisfiable.

If satisfiability is undecidable but semi-decidable, then un*satisfiability cannot be semi-decidable. Hence, hierarchic superposition cannot be refutationally complete for the extended fragment described above (compare with the discussion at the beginning Section 11.3). On the other hand, we have "model completeness" in the sense that, if there is a model, it can be constructed using the procedure we have just described. This is a somewhat unusual situation in automated reasoning, where we often find the opposite: the set of unsatisfiable sentences is semi-decidable, e.g. due to the existence of calculi that are sound and refutationally complete, whereas the set of satisfiable sentences is not recursively enumerable.*[8]

In the remaining subsections we will blend the ideas described above with the encoding from Section 11.2.3. We shall start with an informal description.

11.4.1 Informal Description of the Encoding

Like in Section 11.2, we consider infinite sequences of bits which we divide into chunks, this time of
d a fixed length, determined by an uninterpreted constant symbol d of sort $\mathbb{Q}$. Each of these chunks

[8]The fact that the shift of recursive enumerability from the unsatisfiability problem to the satisfiability problem is a rather peculiar property was brought to the attention of the author of the present thesis by Dietrich Kuske, cf. Footnote 7 on page 279.

has the form

$$\underbrace{001011}_{\substack{\text{left de-}\\\text{limiter}}}\ 1^{\ell}0^{d-\ell-6}\ \underbrace{0011}_{\substack{\text{first sub-}\\\text{delimiter}}}\ 1^{c_1}0^{d-c_1-4}\ \underbrace{0011}_{\substack{\text{second}\\\text{subde-}\\\text{limiter}}}\ 1^{c_2}0^{d-c_2-4}\ ,$$

where ℓ is the address of the program line to be executed, c_1 is the value currently stored in C_1, and c_2 is the value currently stored in C_2. The length of the chunks is $3 \cdot d$ bits each, and every subchunk contains d bits. Obviously, the constant symbol d has to be interpreted by some positive integer that allows the subchunks to store sufficiently large values for the current program line and the current counter values. If $\mathcal{M}$ halts when started on the input $\langle m, n\rangle$, then there exists such a sufficiently large value for d. Due to the length of the used delimiters, we set $\max\{K+6, m+4, n+4\}$ as a lower bound for d, where K is the last line of $\mathcal{M}$'s program. Moreover, the leftmost chunk starts at position 0, i.e. there is a 001011-delimiter starting at position 0 but none starting left of 0.

Figure 11.2 illustrates the structure of a single chunk in the sequence, starting at position x.

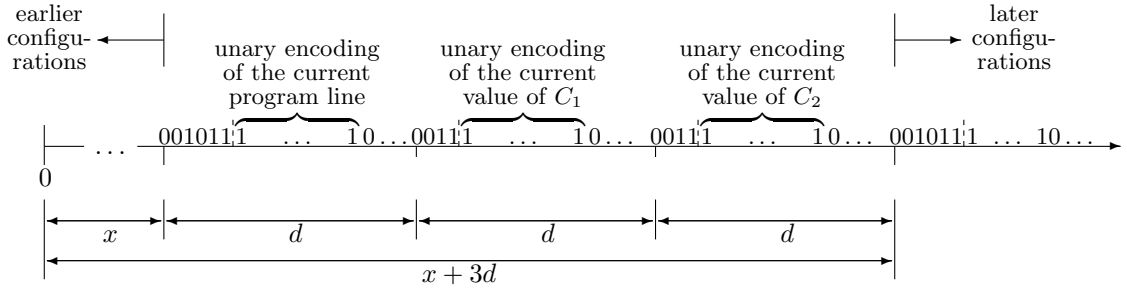


Figure 11.2: Structure of a single chunk of length $3d$.

In contrast to the encoding described in Section 11.2 we this time aim for a finite number of chunks in the sequence, i.e. we only consider finite runs of $\mathcal{M}$. We use the rational number 1 to mark the end of the run: the very last chunk starts at position $1 - 3d$.

For the sake of clarity, we formulate the encoding without restriction to difference constraints in a first step. Then, in Section 11.4.3, we restate the encoding in the more restricted fragment.

11.4.2 Formal Encoding of Two-Counter Machine Computations

Recall that we assume to be given a two-counter machine $\mathcal{M}$ with $K+1$ program lines, labeled $0, \ldots, K$, and two input values m and n. We assume that 0 is the initial program line, that program line K contains the `halt` instruction, and that there is no other program line containing `halt`.

We mark a finite subdomain of $[0, 1]$ which will serve as *pseudo-integers*. The elements of this subdomain shall be uniformly distributed within $[0, 1]$. To this end, we use the uninterpreted unary predicate symbol N which shall contain exactly the pseudo-integers in any model. Moreover, we use an uninterpreted constant c to determine the distance between two pseudo-integers. The following sentence φ_0' sets the stage for using the pseudo-integers $0, c, 2c, 3c, \ldots, 1-2c, 1-c, 1$. N c φ_0'

$$\begin{aligned}\varphi_0' :=\ & c > 0 \\ & \wedge\ N(0)\ \wedge\ N(1)\ \wedge\ \big(\forall x.\, N(x)\ \rightarrow\ 0 \leq x \wedge x \leq 1\big) \\ & \wedge\ \big(\forall xy.\, y - x = c\ \wedge\ N(x)\ \rightarrow\ N(y)\big) \\ & \wedge\ \big(\forall xy.\, x - y < c\ \wedge\ y - x < c\ \wedge\ N(x)\ \wedge\ N(y)\ \rightarrow\ x = y\big) \\ & \wedge\ \big(\forall x.\, P(x)\ \rightarrow\ N(x)\big)\end{aligned}$$

The last subformula ensures that P contains only pseudo-integers. Having distinguished the pseudo-integers from the other rationals in $[0, 1]$, we can now use them as a basis for the encoding.

The used abbreviations ψ_{001011}, ψ_{0011}, etc. have to be adapted as follows:

ψ_{001011} $$\psi_{001011}(t) := \neg P(t) \ \wedge \ \neg P(t+c) \ \wedge \ P(t+2c) \ \wedge \ \neg P(t+3c) \ \wedge \ P(t+4c) \ \wedge \ P(t+5c)$$

ψ_{0011} $$\psi_{0011}(t) := \neg P(t) \ \wedge \ \neg P(t+c) \ \wedge \ P(t+2c) \ \wedge \ P(t+3c)$$

ψ_{01} $$\psi_{01}(t) := \neg P(t) \ \wedge \ P(t+c)$$

ψ_{10} $$\psi_{10}(t) := P(t) \ \wedge \ \neg P(t+c)$$

χ_ℓ $$\chi_\ell(t) := \psi_{10}(t + 5c + \ell \cdot c) \qquad \text{for } \ell = 0, \ldots, K$$

The adapted variants of the sentences $\varphi_1, \ldots, \varphi_4$ are the following. The sentence φ_1' sets up the general structure of the predicate P. Let k again denote the integer $k := \max\{K+6, m+4, n+4\}$.

φ_1'

$$\varphi_1' := \ N(d) \ \wedge \ d \geq k \cdot c \tag{11.9}$$
$$\wedge \ \psi_{001011}(0) \ \wedge \ \psi_{0011}(d) \ \wedge \ \psi_{0011}(2d) \ \wedge \ \psi_{001011}(1-3d) \tag{11.10}$$
$$\wedge \ \big(\forall x.\, \psi_{001011}(x) \ \wedge \ x < 1 - 3d \ \rightarrow \ \psi_{001011}(x+3d)\big) \tag{11.11}$$
$$\wedge \ \big(\forall x.\, \psi_{001011}(x+3d) \ \wedge \ x \geq 0 \ \rightarrow \ \psi_{001011}(x)\big) \tag{11.12}$$
$$\wedge \ \big(\forall x.\, \psi_{0011}(x) \ \wedge \ x < 1 - 3d \ \rightarrow \ \psi_{0011}(x+3d)\big) \tag{11.13}$$
$$\wedge \ \big(\forall x.\, \psi_{0011}(x+3d) \ \wedge \ x \geq 0 \ \rightarrow \ \psi_{0011}(x)\big) \tag{11.14}$$
$$\wedge \ \big(\forall x.\, \psi_{001011}(x) \ \wedge \ x < 3d \ \rightarrow \ x = 0\big) \tag{11.15}$$
$$\wedge \ \big(\forall x.\, \psi_{0011}(x) \ \wedge \ x < 3d \ \rightarrow \ x = 2d \ \vee \ x = d\big) \tag{11.16}$$
$$\wedge \ \big(\forall xy.\, \psi_{001011}(x) \ \wedge \ \psi_{01}(y) \ \wedge \ x + 6c \leq y \ \wedge \ y < x + 3d \ \rightarrow \ \psi_{0011}(y - c)\big) \tag{11.17}$$

Subformula (11.10) sets the first 001011-delimiter at position 0 and the two subchunk delimiters 0011 at positions $d \cdot c$ and $2d \cdot c$. Moreover, the last 001011-delimiter is set to be at position $1 - 3d$. Subformulas (11.11) to Formula (11.14) ensure that there are 001011- and 0011-delimiters evenly distributed between positions 0 and 1. Subformulas (11.15) and (11.17) guarantee that there are no spurious delimiters between 0 and 1. Due to the fact that P contains no elements outside of $[0, 1]$, which is entailed by φ_0', there is no 001011- or 0011-delimiter starting at 0, left of 0, at 1, or right of 1. Subformula (11.17) stipulates that every 01 subsequence is part of one of the delimiters, i.e. there cannot be a subsequence 01 that lies outside of a 001011- or 0011-delimiter. This entails that between one delimiter (001011 or 0011) and the subsequent one there is exactly one subsequence 10, possibly overlapping with the last or first bit of one of the delimiters. Hence, this subsequence uniquely marks the end of the number encoded in the respective subchunk.

The following sentence sets the initial values of the counters. Moreover, it sets the initial program line, which we assume to be the very first one:

$\varphi_2'^{m,n}$ $$\varphi_2'^{m,n} := \ \chi_0(0) \ \wedge \ \psi_{10}(d + 3c + m \cdot c) \ \wedge \ \psi_{10}(2d + 3c + n \cdot c) \ .$$

With the sentence $\varphi_3'^{K}$, we ensure that program lines never exceeds K:

$\varphi_3'^{K}$ $$\varphi_3'^{K} := \ \forall xy.\, \psi_{001011}(x) \ \wedge \ \psi_{10}(y) \ \wedge \ y < x + d \ \rightarrow \ y \leq x + 5c + K \cdot c \ .$$

We also have to encode the condition that the two-counter machine halts at some point in time. Recall that we assume the `halt` instruction to exclusively appear in program line K.

$\varphi_4'^{K}$ $$\varphi_4'^{K} := \ \chi_K(1 - 3d) \ .$$

It remains to encode the control flow of $\mathcal{M}$. We assume that the following instructions occur in program line ℓ for some $\ell \in \{0, \ldots, K\}$.

In the following table we give prototypical encodings of the instructions of two-counter machines. The encoding of operations is only given for counter C_1. The encoding for counter C_2 can be done analogously.

Encoding of the instruction $\ell :$ `inc(`C_1`)`:

$$\forall x.\ \psi_{001011}(x) \ \wedge\ \chi_\ell(x) \ \rightarrow\ \chi_{\ell+1}(x+3d)$$

$$\forall xy.\ \psi_{001011}(x) \ \wedge\ \chi_\ell(x) \ \wedge\ x+d \leq y \ \wedge\ y \leq x+2d \ \wedge\ \psi_{10}(y) \ \rightarrow\ \psi_{10}(y+3d+c)$$

$$\forall xz.\ \psi_{001011}(x) \ \wedge\ \chi_\ell(x) \ \wedge\ x+2d \leq z \ \wedge\ z \leq x+3d \ \wedge\ \psi_{10}(z) \ \rightarrow\ \psi_{10}(z+3d)$$

These three sentences encode a transition from a configuration $\langle \ell, c_1, c_2 \rangle$ to the successor configuration $\langle \ell+1, c_1+1, c_2 \rangle$. The first sentence stipulates that the next program line is the one with the label $\ell+1$. While the second sentence encodes the increase of counter C_1 by 1, the third sentence makes sure that counter C_2 retains its value.

The subfomula $\psi_{001011}(x)$ in the premises of the implications states that the chunk encoding the currently regarded configuration starts at position x. The other preconditions make clear that y and z correspond to the positions at which we find 10 subsequences in the two subchunks storing the current counter values:

$$\begin{array}{lllllll} x & & x+d & y & x+2d & z & \\ \downarrow & & \downarrow & \downarrow & \downarrow & \downarrow & \\ \underbrace{001011}_{\text{left delimiter}} 1^{\ell} 0 \ldots 0 & & \underbrace{0011}_{\text{first subdelimiter}} 1^{c_1-1} 10 \ldots 0 & & \underbrace{0011}_{\text{second subdelimiter}} 1^{c_2-1} 10 \ldots 0 & & \end{array}$$

Hence, C_1 and C_2 currently store the values $c_1 = \frac{1}{c}(y-x-d-3c)$ and $c_2 = \frac{1}{c}(z-x-2d-3c)$, respectively.

Encoding of the instruction $\ell :$ `test&dec(`C_1`,`ℓ'`)`:

The case of C_1 storing 0:

$$\forall x.\ \psi_{001011}(x) \ \wedge\ \chi_\ell(x) \ \wedge\ \psi_{10}(x+d+3c) \ \rightarrow\ \chi_{\ell'}(x+3d)$$

$$\forall x.\ \psi_{001011}(x) \ \wedge\ \chi_\ell(x) \ \wedge\ \psi_{10}(x+d+3c) \ \rightarrow\ \psi_{10}(x+4d+3c)$$

$$\forall xz.\ \psi_{001011}(x) \ \wedge\ \chi_\ell(x) \ \wedge\ \psi_{10}(x+d+3c) \ \wedge\ x+2d \leq z \ \wedge\ z \leq x+3d \ \wedge\ \psi_{10}(z) \ \rightarrow\ \psi_{10}(z+3d)$$

The condition $\psi_{10}(x+d+3c)$ ensures that the first counter stores the value 0.

The case of C_1 storing a value greater than 0:

$$\forall xy.\ \psi_{001011}(x) \ \wedge\ \chi_\ell(x) \ \wedge\ x+d+3c < y \ \wedge\ y \leq x+2d \ \wedge\ \psi_{10}(y) \ \rightarrow\ \chi_{\ell+1}(x+3d)$$

$$\forall xy.\ \psi_{001011}(x) \ \wedge\ \chi_\ell(x) \ \wedge\ x+d+3c < y \ \wedge\ y \leq x+2d \ \wedge\ \psi_{10}(y) \ \rightarrow\ \psi_{10}(y+3d-c)$$

$$\forall xz.\ \psi_{001011}(x) \ \wedge\ \chi_\ell(x) \ \wedge\ \neg\psi_{10}(x+d+3c) \ \wedge\ x+2d \leq z \ \wedge\ z \leq x+3d \ \wedge\ \psi_{10}(z) \ \rightarrow\ \psi_{10}(z+3d)$$

The condition $y > x+d+3$ ensures that the first counter stores a value strictly greater than 0. The same applies to condition $\neg\psi_{10}(x+d+3c)$ in the third sentence.

Encoding of the instruction $\ell :$ `goto(`ℓ'`)`:

$$\forall x.\ \psi_{001011}(x) \ \wedge\ \chi_\ell(x) \ \rightarrow\ \chi_{\ell'}(x+3d)$$

$$\forall xy.\ \psi_{001011}(x) \ \wedge\ \chi_\ell(x) \ \wedge\ x+d+3c \leq y \ \wedge\ y \leq x+2d \ \wedge\ \psi_{10}(y) \ \rightarrow\ \psi_{10}(y+3d)$$

$$\forall xz.\ \psi_{001011}(x) \ \wedge\ \chi_\ell(x) \ \wedge\ x+2d \leq z \ \wedge\ z \leq x+3d \ \wedge\ \psi_{10}(z) \ \rightarrow\ \psi_{10}(z+3d)$$

Encoding of the instruction K : `halt`:

$$\forall x.\ \psi_{001011}(x) \ \wedge\ \chi_K(x) \ \rightarrow\ x = 1 - 3d$$

The sentence stipulates that, if program line K is reached, then only at the end of the computation.

Lemma 11.4.2. *Let $\mathcal{M}$ be any two-counter machine and let $\varphi'_{\mathcal{M}}$ be the encoding of its behavior as described above. Then, $\mathcal{M}$ halts on the input $\langle m, n\rangle$ if and only if there is a model $\mathcal{A}$ of the set of* LRA+PN *sentence $\varphi'_1 \wedge \ldots \wedge {\varphi'_4}^K \wedge \varphi_{\mathcal{M}}$. Moreover, in that case, $P^{\mathcal{A}}$ and $N_{\mathcal{A}}$ are finite subsets of the rational unit interval $[0, 1]$.*

Theorem 11.4.3. *Satisfiability for the $\exists^2\forall^2$ fragment of* LRA+PN *is undecidable, even if the arithmetic domain is the rational interval $[0, 1]$ and the interpretations of P and N are restricted to finite subsets of $[0, 1]$.*

11.4.3 Restriction to Difference Constraints

As the last step, we now adapt the encoding from the previous section to exclusively use difference constraints $v - v' \triangleleft c'$ with $\triangleleft \in \{<, \leq, =, \neq, \geq, >\}$, where c' is either a rational number or an uninterpreted constant symbol of sort $\mathbb{Q}$. In addition, we allow the arithmetic atom $x_0 = 0$.

The sentence φ'_0 requires only small changes, which yield

φ'_0

$$\begin{aligned}\varphi'_0 :=\ & \big(\forall x.\, x - x < c\big)\\ & \wedge\ \big(\forall x_0 x_1.\, x_0 = 0 \ \wedge\ x_1 - x_0 = 1 \ \rightarrow\ N(x_0) \ \wedge\ N(x_1)\big)\\ & \wedge\ \big(\forall x x_0.\, x_0 = 0 \wedge N(x) \ \rightarrow\ 0 \leq x - x_0 \wedge x - x_0 \leq 1\big)\\ & \wedge\ \big(\forall x y.\, y - x = c \ \wedge\ N(x) \ \rightarrow\ N(y)\big)\\ & \wedge\ \big(\forall x y.\, x - y < c \ \wedge\ y - x < c \ \wedge\ N(x) \ \wedge\ N(y) \ \rightarrow\ x - y = 0\big)\\ & \wedge\ \big(\forall x.\, P(x) \ \rightarrow\ N(x)\big)\ .\end{aligned}$$

In the rest of the encoding, the abbreviations ψ_{001011}, ψ_{0011}, etc. cannot be used this comfortably anymore, as the following reformulation of φ'_1 shows.

The subformula $d \geq k \cdot c$ is replaced with

$$\forall x_1 \ldots x_{k+1}.\, x_2 - x_1 = c \ \wedge\ x_3 - x_2 = c \ \wedge \ldots \wedge\ x_k - x_{k-1} = c \ \wedge\ x_{k+1} - x_k = c \ \rightarrow\ x_{k+1} - x_1 \leq d\ .$$

The subformula $\psi_{001011}(0)$ is replaced with

$$\forall x_0 x_1 \ldots x_5.\, x_0 = 0 \ \wedge\ x_1 - x_0 = c \ \wedge\ x_2 - x_1 = c \ \wedge \ldots \wedge\ x_5 - x_4 = c \ \rightarrow\ \neg P(x_0) \ \wedge\ \neg P(x_1) \ \wedge\ P(x_2) \ \wedge\ \neg P(x_3) \ \wedge\ P(x_4) \ \wedge\ P(x_5)\ .$$

The subformula $\psi_{0011}(d)$ is replaced with

$$\forall x_0 x_1 \ldots x_4.\, x_0 = 0 \ \wedge\ x_1 - x_0 = d \ \wedge\ x_2 - x_1 = c \ \wedge\ x_3 - x_2 = c \ \wedge\ x_4 - x_3 = c \ \rightarrow\ \neg P(x_1) \ \wedge\ \neg P(x_2) \ \wedge\ P(x_3) \ \wedge\ P(x_4)\ .$$

The subformula $\psi_{0011}(2d)$ is replaced with

$$\forall x_0 x_1 \ldots x_4.\, x_0 = 0 \ \wedge\ x_1 - x_0 = d \ \wedge\ x_2 - x_1 = d \ \wedge\ x_3 - x_2 = c \ \wedge\ x_4 - x_3 = c \ \wedge\ x_5 - x_4 = c \ \rightarrow\ \neg P(x_2) \ \wedge\ \neg P(x_3) \ \wedge\ P(x_4) \ \wedge\ P(x_5)\ .$$

The subformula $\psi_{001011}(1 - 3d)$ is replaced with

$$\begin{aligned}\forall x_0 x_1 \ldots x_8.\, \big(& x_0 = 0\\ & \wedge\ x_1 - x_0 = 1 \ \wedge\ x_1 - x_2 = d \ \wedge\ x_2 - x_3 = d \ \wedge\ x_3 - x_4 = d\\ & \wedge\ x_5 - x_4 = c \ \wedge\ x_6 - x_5 = c \ \wedge\ x_7 - x_6 = c \ \wedge\ x_8 - x_7 = c \ \wedge\ \ \wedge\ x_9 - x_8 = c\big)\\ & \rightarrow\ \neg P(x_4) \ \wedge\ \neg P(x_5) \ \wedge\ P(x_6) \ \wedge\ \neg P(x_7) \ \wedge\ P(x_8) \ \wedge\ P(x_9)\ .\end{aligned}$$

The subformula $\big(\forall x.\ \psi_{001011}(x)\ \wedge\ x < 3d\ \rightarrow\ x = 0\big)$ is replaced with

$$\begin{aligned}\forall x_0 x_1 \ldots x_6 y_1 y_2 y_3.\ \big(&x_0 = 0\\ &\wedge\ y_1 - x_0 = d\ \wedge\ y_2 - y_1 = d\ \wedge\ y_3 - y_2 = d\\ &\wedge\ x_2 - x_1 = c\ \wedge\ x_3 - x_2 = c\ \wedge\ x_4 - x_3 = c\ \wedge\ x_5 - x_4 = c\ \wedge\ \ x_6 - x_5 = c\\ &\wedge\ \neg P(x_1)\ \wedge\ \neg P(x_2)\ \wedge\ P(x_3)\ \wedge\ \neg P(x_4)\ \wedge\ P(x_5)\ \wedge\ P(x_6)\ \wedge\ x_1 - y_3 < 0\big)\\ &\quad\rightarrow\ \ x_1 - x_0 = 0\ .\end{aligned}$$

The other constituents of φ_1' and the sentences ${\varphi_2'}^{m,n}, \ldots, {\varphi_4'}^{K}$ can be modified in the same spirit.

Regarding the encoding of $\mathcal{M}$'s control flow, we show two examples of how to modify the respective sentences.

Encoding of the instruction ℓ : `inc(`C_1`)`:

The subformula

$$\forall xy.\ \psi_{001011}(x)\ \wedge\ \chi_\ell(x)\ \wedge\ x + d \leq y\ \wedge\ y \leq x + 2d\ \wedge\ \psi_{10}(y)\ \ \rightarrow\ \ \psi_{10}(y + 3d + c)$$

is replaced with

$$\begin{aligned}\forall xy x_1 \ldots x_5 v_1 \ldots v_{\ell+1} y_1 y_2 u_1 u_2 u_3 u_4.&\\ \big(\ &x_1 - x = c\ \wedge\ x_2 - x_1 = c\ \wedge \ldots \wedge\ x_5 - x_4 = c\\ &\wedge\ \neg P(x)\ \wedge\ \neg P(x_1)\ \wedge\ P(x_2)\ \wedge\ \neg P(x_3)\ \wedge\ P(x_4)\ \wedge\ P(x_5)\\ &\wedge\ v_1 - x_5 = c\ \wedge\ v_2 - v_1 = c\ \wedge \ldots \wedge\ v_{\ell+1} - v_\ell = c\ \wedge\ P(v_\ell)\ \wedge\ \neg P(v_{\ell+1})\\ &\wedge\ y_1 - x = d\ \wedge\ y_1 - y \leq 0\ \wedge\ y - y_1 \leq d\\ &\wedge\ y_2 - y = c\ \wedge\ P(y)\ \wedge\ \neg P(y_2)\\ &\wedge\ u_1 - y = d\ \wedge\ u_2 - u_1 = d\ \wedge\ u_3 - u_2 = d\ \wedge\ u_4 - u_3 = c\ \wedge\ u_5 - u_4 = c\big)\\ &\quad\rightarrow\ \ P(u_4)\ \wedge\ \neg P(u_5)\ .\end{aligned}$$

Encoding of the instruction K : `halt`: The sentence $\forall x.\ \psi_{001011}(x)\ \wedge\ \chi_K(x)\ \ \rightarrow\ \ x = 1 - 3d$ is replaced with

$$\begin{aligned}\forall x x_0 x_1 \ldots x_5 v_1 \ldots v_{K+1} z_1 z_2 z_3 z_4.&\\ \big(\ &x_1 - x = c\ \wedge\ x_2 - x_1 = c\ \wedge \ldots \wedge\ x_5 - x_4 = c\\ &\wedge\ \neg P(x)\ \wedge\ \neg P(x_1)\ \wedge\ P(x_2)\ \wedge\ \neg P(x_3)\ \wedge\ P(x_4)\ \wedge\ P(x_5)\\ &\wedge\ v_1 - x_5 = c\ \wedge\ v_2 - v_1 = c\ \wedge \ldots \wedge\ v_{K+1} - v_K = c\big)\ \wedge\ P(v_K)\ \wedge\ \neg P(v_{K+1})\\ &\wedge\ x_0 = 0\ \wedge\ z_1 - x_0 = 1\ \wedge\ z_1 - z_2 = d\ \wedge\ z_2 - z_3 = d\ \wedge\ z_3 - z_4 = d\big)\\ &\quad\rightarrow\ \ x - z_4 = 0\ .\end{aligned}$$

Theorem 11.4.4. *Satisfiability for the* LRA+PN *is undecidable, even if arithmetic atoms are restricted to difference constraints plus atoms* $x_0 = 0$*, the arithmetic domain is the rational or real interval* $[0, 1]$*, and the interpretations of* P *and* N *are restricted to finite subsets of* $[0, 1]$*.*

11.5 Relevance to Verification

Verification of hardware and software is one driving force behind attempts to the combination of theories, such as integer or real arithmetic and the *theory of equality over uninterpreted functions (EUF)* — EUF is understood to refer to the collection of all logical Σ-theories containing all valid quantifier-free Σ-sentences over a finite vocabulary Σ without predicate symbols.[9] For quantifier-free cases the Nelson–Oppen framework provides a general-purpose approach for the

EUF

[9] In the literature, the definition of EUF often includes uninterpreted predicate symbols, which are then ignored in the further treatment for convenience. See, e.g., Section 3.2 in [BM07], or Section 4.2 in [KS16].

construction of decision procedures (see Section 10.3). Over the course of the last fifteen year numerous approaches have been proposed to go beyond the quantifier-free setting and handle quantification, see e.g. [FJS04, DNS05, GdM09, GBT09, BMR13, RTdM14, RK15, RBF18, Bar17]. Typically, some kind of heuristic is applied to guide instantiation towards equisatisfiable formulas that are quantifier free. Often the methods are incomplete in the sense that unsatisfiable sentences are not necessarily recognized as such. Nevertheless, the proposed methods have been implemented and successfully applied, e.g. in the tools *Verifun*, *Simplify*, and the *CVC* family.

In verification one usually abstracts from some of the limitations that apply to real-world computing devices. In particular, memory is often regarded as an inexhaustible resource in one way or another. This can take the form of infinitely many memory locations — similar to the infinite tape of a Turing machine — or the form of the capability of storing arbitrarily large integers in single memory location — similar to the counters of counter machines. In our encoding of two-counter machines in Sections 11.2 and 11.4 the uninterpreted predicate symbol P serves as a representation of an unbounded memory. As we have pointed out, any interpretation $P^{\mathcal{A}} \subseteq \mathbb{N}$ can be conceived as an infinite sequence of bits. And these bits can be accessed by integer addresses. We have also pointed out in Section 11.2.6 that the same applies to uninterpreted function symbols over the integers or some co-domain with at least two distinct elements. This means that our results are relevant to all verification approaches in which an infinite memory is modeled and in which there are sufficiently strong means available to access individual memory locations. Such approaches inevitably face undecidability when they allow too liberal syntax. We shall discuss several exemplary settings: separation logic over an integer-indexed heap, logics formalizing integer-indexed arrays or similar data structures, logics with restricted forms of linear integer arithmetic. We shall also give reasons why incomplete heuristics is sometimes the best one could hope for.

11.5.1 Separation Logic

In [RIS17] the Bernays–Schönfinkel–Ramsey fragment ($\exists^*\forall^*$-sentences) of separation logic is investigated. The quantifiers range over memory locations. Although the authors also present a refinement of Halpern's undecidability result [Hal91] for PA+P, their approach differs from our approach in Section 11.2 in an important aspect. In their setting it is sufficient to consider models in which the unary predicate symbol P is interpreted with a *finite* subset of $\mathbb{N}$. In our setting in Section 11.2 finite subsets do not suffice. It is due to this difference, that their strategy can be used to also show undecidability of the satisfiability problem for $\exists^*\forall^*$-sentences of separation logic over a heap with *finitely* many integer-indexed memory locations, each capable of storing one integer of arbitrary size.

Our results in Sections 11.2 and 11.3 have implications for settings with integer-indexed heaps that comprise a countably infinite number of memory locations, each capable of distinguishing at least two values (e.g. 0 and 1) or states (e.g. *allocated* and *not allocated*). However, a slight modification of the encoding in Section 11.2.2 leads to a result that subsumes Theorem 3 in [RIS17] and also entails undecidability of the satisfiability problem for the $\exists^*\forall^*$-fragment of separation logic with integer-indexed heaps that comprise *only finitely many* memory locations, each capable of storing at least one bit of information.

Lemma 11.5.1. *Let $\mathcal{M}$ be a two-counter machine with $K+1$ program lines, labeled $0, \ldots, K$, and let $\langle m, n\rangle$ be a pair of nonnegative integers. There is a sentence φ from the $(\exists\forall^*)$-fragment of* PA+P*, such that the following statements are equivalent:*

(a) φ is satisfied by a model $\mathcal{A}$ under which $P^{\mathcal{A}}$ is a finite subset of $\mathbb{N}$,

(b) $\mathcal{M}$ reaches the `halt` *instruction when started on the input $\langle m, n\rangle$.*

Proof sketch. The following is a blend of ideas from Sections 11.2 and 11.4. Let $\varphi''_{\mathcal{M}}$ be the encoding of $\mathcal{M}$'s program in accordance with Section 11.2.2 with the exception that we do not encode the instruction in program line K. Due to our conventions, this program line contains the `halt` instruction. Let $\varphi''_1(z)$ result from φ_1 after replacing the Subformula (11.3) with

$$\forall x.\, x < z \;\wedge\; \psi_{001011}(x) \;\longrightarrow\; \psi_{0011}(2x) \;\wedge\; \psi_{0011}(3x) \;\wedge\; \psi_{001011}(4x)\ .$$

Moreover, let

$$\varphi_4''^K(z) := \ \psi_{001011}(z) \wedge \chi_K(z) \ .$$

Notice that both formulas $\varphi_1''(z)$ and $\varphi_4''(z)$ contain the free variable z. We now set

$$\varphi := \ \exists z.\ \varphi_1''(z) \wedge \varphi_2^{m,n} \wedge \varphi_3^K \wedge \varphi_{\mathcal{M}}'' \wedge \varphi_4''^K(z) \ .$$

There exists a model $\mathcal{A}$ of φ if and only if $\mathcal{M}$ reaches program line K when started on the input $\langle m, n\rangle$. Due to the modifications in φ_1'', the formula $\psi_{001011}(x)$ does not have to be satisfied for arbitrarily large values of x. One consequence is that the run of $\mathcal{M}$ represented by a model of φ can be aborted at the point when program line K is reached. This means, in contrast to the proof of Lemma 11.2.1, we do not have to artificially continue $\mathcal{M}$'s run beyond that point. Hence, any model of φ can be modified in such a way that from a certain point on the bit sequence represented by the interpretation of P contains only zeros. □

11.5.2 Verification of Data Structures

There are undecidability results in the context of verification of programs that use integer-indexed arrays as data structures. Examples can be found in [BMS06] (Section 5), [Bra07] (Sections 2.4 and 2.6.3), [HIV08] (Section 3). The reductions presented therein are based on arrays with infinite co-domains, such as the integers or the reals. Moreover, they typically use at least one quantifier alternation (but face other restrictions of syntax). Usually, several arrays are used for convenience, but could be merged into one. For our proof approach a single array is sufficient as well.

Read operations on integer-indexed arrays can be formalized as uninterpreted function symbols with an integer domain. Hence, our results, Theorems 11.2.7 and 11.2.8 in particular, show that reasoning about integer- or real-indexed arrays over a *finite co-domain* with at least two elements can lead to undecidability, if constraints on array indices provide the necessary syntactic means. Notice that for the proof it is not necessary to have write operations on arrays. This means, a single integer-indexed read-only array over a Boolean co-domain suffices.

The mentioned results and arguments hold for arrays that comprise an infinite number of elements. However, due to Lemma 11.5.1, undecidability arises also in the context of finite arrays (over finite co-domains), as long as their length is not bounded by a concrete number.

Remark 11.5.2. *The above arguments are also applicable to recursively defined data structures, such as lists or trees, as soon as there are sufficiently strong syntactic means available to access the stored information. That is, if one can essentially simulate arrays using a recursive data structure, then our results apply immediately. Examples of such setting are lists where the stored elements can be addressed by integers, or where one can access the sublist starting at the position that is x nodes away from the head (for some integer-sort variable x for which universal quantification is admitted).*

11.5.3 Verification Using Counter Arithmetic

In [BLS02] the fragment *CLU* is introduced, which constitutes a strongly restricted fragment of Presburger arithmetic with additional uninterpreted function and predicate symbols. A less syntactically sugared subfragment is treated in [GHN+04] and in [ABRS09]. There are only two arithmetic operators available in CLU: the successor operator `succ` and the predecessor operator `pred`. There is no interpreted constant symbol available addressing zero or any other concrete integer. On the other hand, some syntactic elements are added for convenience, such as lambda abstraction and an `if-then-else` operator. The fragment was chosen for its expressiveness and the fact that it facilitates efficient reasoning. Although quantifier-free in its original definition, the authors state about their verification tool *UCLID* that they "have built some support for quantifiers in CLU using automatic quantifier instantiation heuristics" ([BLS02], Section 7).

In what follows, we consider the extension of CLU with universal quantification for integer variables. We shall refer to this extended language as *uCLU*. By a result due to Gurevich [Gur76]

(see also [BGG97], Theorems 4.1.8 and 4.1.11), satisfiability of EUF sentences with universal quantification is undecidable. Hence, satisfiability of uCLU sentences is undecidable as well.

Proposition 11.5.3 (Corollary of the Main Theorem in [Gur76])**.** *(Un)satisfiability for uCLU sentences is undecidable.*

On the other hand, the unsatisfiable sentences of first-order logic without interpreted symbols (and thus also of quantified EUF) are recursively enumerable. We next argue that uCLU does not possess this property.

The encoding of two-counter machines from Section 11.2 and 11.3 cannot immediately be translated into uCLU. First of all, we need to fix a point of reference that serves as zero (CLU does not contain 0 as a built-in constant). Moreover, expressions of the form $k \cdot x$ for any integer k and any integer-sort variable x require a form of addition that is not available as a built-in operation in uCLU. However, with unrestricted universal quantification over integer variables at hand, we can easily define addition as a function. Hence, we only need the following uninterpreted symbols to encode two-counter machines: one constant symbol c_0 serving as zero, one binary function symbol realizing addition, one uninterpreted unary function or predicate symbol serving as memory.

We define the addition function (on nonnegative integers) as follows, where we use c_0 as zero:

$$\begin{array}{lll} \forall x. & & \mathrm{add}(x, c_0) = x \\ \forall xy. & \mathrm{succ}(y) > c_0 \;\longrightarrow & \mathrm{add}\big(x, \mathrm{succ}(y)\big) = \mathrm{add}\big(\mathrm{succ}(x), y\big) \\ \forall xy. & \mathrm{succ}(y) < c_0 \;\longrightarrow & \mathrm{add}\big(x, \mathrm{succ}(y)\big) = x \ . \end{array}$$

All abbreviations $k \cdot x$ are unfolded into $\mathrm{add}(x, \mathrm{add}(x, \ldots \mathrm{add}(x, x) \ldots))$ and all integers that we have used in the encoding from Section 11.2.3 shall be written as $\mathrm{succ}^k(c_0) := \mathrm{succ}(\ldots \mathrm{succ}(c_0) \ldots)$ instead of just k. Moreover, we add guards $x \geq c_0 \rightarrow \ldots$ to each sentence for every universally quantified variable x that occurs in that sentence.

As we have seen in Section 11.3, in particular in Theorems 11.3.7 and 11.3.9, $\forall\exists$ quantifier alternations yield (un)satisfiability problems that are not even recursively enumerable. Since CLU allows uninterpreted function symbols, uCLU essentially allows $\forall^*\exists^*$ quantifier prefixes (modulo Skolemization). Hence, we may introduce a fresh unary Skolem function f_{init} and translate the sentence φ_5' from Section 11.3 into the uCLU formula

$$\forall x.\, x \geq 0 \;\longrightarrow\; x \leq f_{\mathrm{init}}(x) \;\wedge\; \psi_{001011}\big(f_{\mathrm{init}}(x)\big) \;\wedge\; \chi_0\big(f_{\mathrm{init}}(x)\big) \ .$$

This means, we can transfer Theorem 11.3.9 to uCLU and thus obtain the following result.

Theorem 11.5.4. *Neither the set of satisfiable uCLU sentences nor the set of unsatisfiable uCLU sentences is recursively enumerable. In particular, there cannot be any sound and refutationally complete calculus for uCLU.*

In [ABRS09] the authors present a combination result (Theorem 4.6) for the ground theories of *integer-offsets* (the arithmetic subfragment of CLU embodied by the operators `succ` and `pred`), arrays, and/or EUF (as long as the signature of uninterpreted functions does not contain the array sort). The result states that the satisfiability of sentences in such combined theories can be decided using term-rewriting methods. By a similar line of argument that led us to Proposition 11.5.4, it follows that Theorem 4.6 in [ABRS09] cannot be generalized to cases which admit quantification over integer-sort variables. But we do not only lose decidability, we also lose semi-decidability. In other words, it is impossible to devise sound and complete calculi for combinations of EUF and arithmetic — even in such a restricted form as in CLU — if universal quantification of integer variables is admitted.

11.5.4 Almost Uninterpreted Formulas with Offsets

almost uninterpreted fragment

In [GdM09] Ge and de Moura define the fragment of *almost uninterpreted formulas*. It constitutes a combination of subfragments of first-order logic, EUF, and linear arithmetic over the integers.

Its language admits uninterpreted predicate symbols, function symbols, and constant symbols. Formulas are assumed to be given in CNF. All occurring variables are universally quantified, but may only occur as arguments of uninterpreted function or predicate symbols with the following exceptions. Literals of the form $\neg(x \leq y)$, $\neg(x \leq t)$, $\neg(x \geq t)$, $\neg(x = t)$, $\neg(x \leq y + t)$, $x = t$ with variables x, y of sort $\mathbb{Q}$ are allowed for all ground terms t of the integer sort. Moreover, terms of the form $f(\ldots, x + t, \ldots)$ and $P(\ldots, x + t, \ldots)$ are allowed for ground terms t of the integer sort, uninterpreted function symbols f and uninterpreted predicate symbols P. In what follows we shall be more liberal with the syntax than this. However, the formulas that we will present can be rewritten into equivalent ones that obey the above restrictions. Consequently, we will be able to show undecidability of the associated satisfiability problem.

The encoding of two-counter machines given in Section 11.2 requires different syntactic means than the ones available in Ge and de Moura's almost uninterpreted fragment. Hence, a proof of undecidability in the syntax of [GdM09] needs a slight shift of paradigm similar to the one described in Section 11.4.1. We start from the encoding presented in Section 11.2.3, since it requires at most two integer-sort variables in arithmetic atoms. The length of the chunks storing a single configuration $\langle \ell, c_1, c_2 \rangle$ increases over time. This behavior is necessary to formalize non-terminating runs — and recurring runs in particular — by satisfiable formulas. However, in order to formalize a run that eventually reaches the `halt` instruction by a satisfiable sentence, it suffices to fix the length of the chunks representing a single configuration to a size that can accommodate all configurations that occur in the run, depending on the machine program and on the given input. In Ge and de Moura's fragment uninterpreted constant symbols are available that can be used for this purpose. In what follows, the uninterpreted constant d is used to determine the length of subchunks, as depicted in Figure 11.3. Moreover, we now start the encoding of the run at the very first bit of the

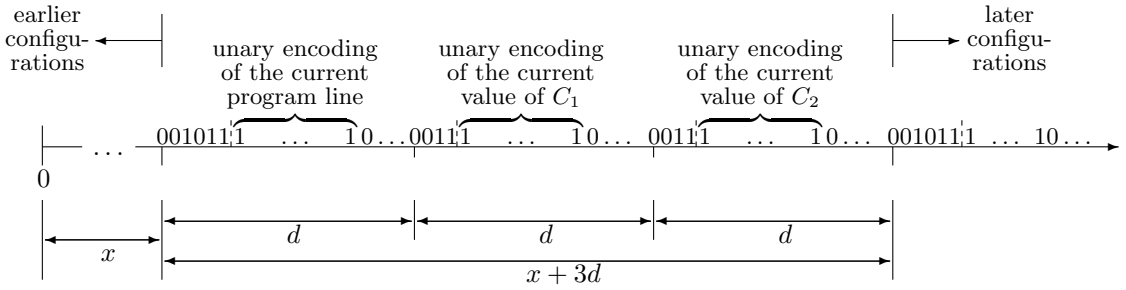


Figure 11.3: Structure of a single chunk of constant length $3d$.

bit sequence represented by P. We replace the sentence φ_1 (page 269) with the following sentence φ_1'''. Let k be the result of the expression $\max(K + 6, m + 4, n + 4)$, where K is the address of the last program line and m and n are the input values. The purpose of the uninterpreted constant e is to mark the end of the run, as we will see later.

$\varphi_1''' :=$ φ_1'''

$$
\begin{aligned}
& d \geq k \ \wedge\ e \geq 0 \ \wedge\ \big(\forall x.\, x \leq -1 \ \rightarrow\ \neg P(x)\big) \ \wedge\ \big(\forall x.\, x \geq e + 3d \ \rightarrow\ \neg P(x)\big) \\
& \wedge\ \psi_{001011}(0) \ \wedge\ \psi_{001011}(e) \\
& \wedge\ \big(\forall x.\, \psi_{001011}(x) \ \wedge\ x \leq e - 1 \ \rightarrow\ \psi_{001011}(x + 3d)\big) \\
& \wedge\ \big(\forall x.\, \psi_{001011}(x) \ \wedge\ x \leq e \ \rightarrow\ \psi_{0011}(x + d) \ \wedge\ \psi_{0011}(x + 2d)\big) \\
& \wedge\ \big(\forall xy.\, \psi_{001011}(x) \ \wedge\ \psi_{001011}(y) \ \wedge\ x \leq y - 1 \ \wedge\ y \leq x + 3d - 1 \ \rightarrow\ \texttt{false}\big) \\
& \wedge\ \big(\forall xy.\, \psi_{001011}(x) \ \wedge\ \psi_{0011}(y) \ \wedge\ x \leq y - 1 \ \wedge\ y \leq x + d - 1 \ \rightarrow\ \texttt{false}\big) \\
& \wedge\ \big(\forall xyz.\, \psi_{001011}(x) \wedge \psi_{0011}(y) \wedge \psi_{0011}(z) \wedge x \leq y - 1 \wedge y \leq z - 1 \wedge z \leq x + 2d - 1 \rightarrow \texttt{false}\big) \\
& \wedge\ \big(\forall xy.\, \psi_{001011}(x) \ \wedge\ \psi_{01}(y) \ \wedge\ x \leq y - 6 \ \wedge\ y \leq x + 3d - 1 \ \rightarrow\ \psi_{0011}(y - 1)\big)
\end{aligned}
$$

The sentences $\varphi_2^{m,n}$ and φ_3^K (page 270) can be adapted in the same spirit:

$\varphi_2'''^{m,n}$ $\varphi_2'''^{m,n} := \chi_0(0) \wedge \psi_{10}(d+3+m) \wedge \psi_{10}(2d+3+n)$

$\varphi_3'''^{K}$ $\varphi_3'''^{K} := \forall xy.\, \psi_{001011}(x) \wedge \psi_{10}(y) \wedge x \leq y - (5+K+1) \wedge y \leq x + d \longrightarrow \texttt{false}\,.$

The adapted encoding of an instruction $\ell : \texttt{inc}(C_1)$ comprises the formulas

$$\forall xy.\, \psi_{001011}(x) \wedge x \leq y - d \wedge y \leq x + 2d \wedge \psi_{10}(y) \wedge \chi_\ell(x) \longrightarrow \psi_{10}(y+3d+1) \wedge \chi_{\ell+1}(x+3d)$$

$$\forall xz.\, \psi_{001011}(x) \wedge x \leq z - 2d \wedge z \leq x + 3d \wedge \psi_{10}(z) \wedge \chi_\ell(x) \longrightarrow \psi_{10}(z+3d)$$

The other instructions can be adapted analogously. The only exception is the `halt` instruction in the last program line which we shall not encode, as in the proof sketch for Lemma 11.5.1.

Finally, we also have to modify the condition that the two-counter machine halts at some point in time. We use the uninterpreted constant e for this purpose:

$\varphi_4'''^{K}$ $$\varphi_4'''^{K} := \chi_K(e)\,.$$

Consequently, using the fragment given in [GdM09], we can encode the halting problem of a two-counter machine $\mathcal{M}$ on input $\langle m, n\rangle$ using only a single uninterpreted unary predicate symbol P (or a single function symbol) plus two uninterpreted constant symbols d, e. More precisely, if $\mathcal{M}$ halts on $\langle m, n\rangle$, then there is model $\mathcal{A}$ of the encoding sentence such that $P^{\mathcal{A}}$ is a finite set of integers.

Theorem 11.5.5. *The satisfiability for the almost uninterpreted fragment with integer offsets is undecidable.*

The outlined encoding is sufficient for a halting run of a two-counter machine. However, we cannot encode recurring counter machines in this way. Thus, we do not obtain hardness beyond recursive enumerability. Indeed, this is in line with [GdM09], where a refutationally complete calculus is given for the described fragment.

The realm of recursive enumerability can be left easily. For instance, it is sufficient to allow scalar multiplication combined with addition for integer-sort variables, i.e. expressions of the form $2 \cdot x + y$. With this construct, we could encode a progressively increasing chunk length. Moreover, uninterpreted function symbols of positive arity can be used to simulate $\forall\exists$ quantifier alternations. Similarly, it would suffice to admit expressions $g(x) + 2$, as we can define, e.g.,

$$\mathrm{times}_c(0) = 0 \;\wedge\; \forall x.\, x \geq 0 \;\rightarrow\; \mathrm{times}_k(x+1) = \mathrm{times}_c(x) + 2$$

for any positive integer k. With a syntax extended this way, one could realize the encoding from Section 11.2.2.

Chapter 12

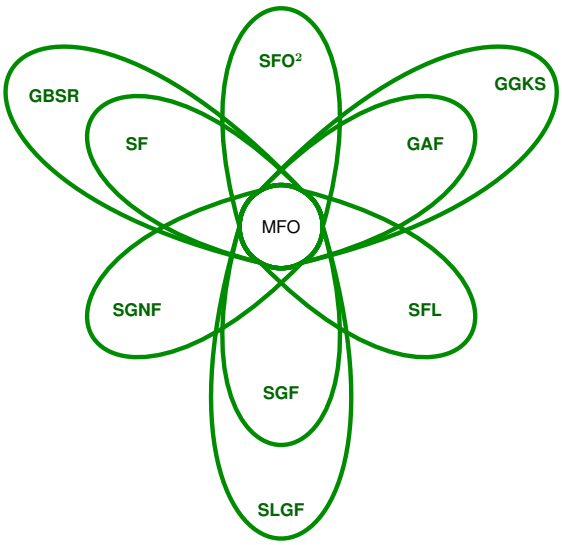
Conclusion

12.1 Separateness of First-Order Variables: Applications to the Classical Decision Problem and Other Areas

In Part I of the present thesis we have introduced the concept of separateness of (sets of) first-order variables and have examined its potential in the context of the classical decision problem and beyond. Although the notion is easy to state and grasp, it opens the door to a number of discoveries. We have mainly concentrated on decidable fragments of first-order logic and have only briefly touched other areas of application. One important property of separateness is its being orthogonal to the syntactic properties that characterize many of the known decidable fragments. Even much better, separateness turned out to be an enabler for the definition of significant syntactic extensions of at least nine such fragments. The reason is that suitable conditions based on separateness of first-order variables often allow for more subtlety when formulating syntactic restrictions, which in the end yields relaxed syntactic conditions. Figure 12.1 depicts once again the novel fragments that we have defined and investigated in Chapter 3 (compare also Figure 1 on page 3). Hence, separateness opens a new perspective on the landscape that research activity around the classical decision problem has revealed over the course of the last about one hundred years. It seems likely that separateness could be used to extend more decidable first-order fragments. For instance, the Skolem fragment and Maslov's fragment K may be interesting candidates for being extended, as may be the more recent unary-negation fragment and the uniform one-dimensional fragment.

Interestingly, each and every of the novel fragments discussed in Chapter 3 properly contains MFO. The reason is simply that in MFO sentences, by definition, any two disjoint sets of first-order variables are separated. The inclusion of MFO could be conceived as a litmus test concerning the generality of definitions of first-order fragments based on separateness: if MFO is not covered, then the definition is not yet liberal enough. That is to say that, if the definition of a first-order fragment is sufficiently strongly based on separateness, then it will inevitably contain MFO.

Another peculiarity is that every extended fragment exhibits the same expressiveness as the underlying original fragment does, but only at the qualitative level. More precisely, we have devised a translation procedure for every extended fragment, say F, which is capable of transforming any given sentence based on the extended syntax of F into an equivalent sentence that belongs to the original fragment, say G. From this perspective, the syntax of G could be conceived as a kind of normal form with respect to F: there is a procedure bringing any F-sentence into G-normal form, so to speak. Furthermore, we have seen that this translation for several extended fragments inevitably leads to a super-polynomial blowup of the formula length in the worst case — see Table 12.1 for an overview. For the translations SF-to-BSR, GBSR-to-BSR, SGF-to-GF, and SLGF-to-LGF, the incurred cost is even so large that it cannot be bounded using elementary functions alone. This shows that the extension of decidable first-order fragments using separateness of variables provides the ability to express certain logical properties in a significantly more succinct way, much rather than yielding any qualitative improvement regarding expressive power. The presented derivations



MFO – monadic first-order fragment
SF – separated fragment
GBSR – generalized BSR
SFO^2 – separated FO^2
GAF – generalized AF
GGKS – generalized GKS
SFL – separated FL
SGF – separated GF
SLGF – separated LGF
SGNFO – separated GNFO

Figure 12.1: Schematic overview of the extended fragments (in green) that have been presented in Chapter 3. Notice that MFO is properly contained in all extended fragments. The focus is on the overlaps with MFO and on the proper containment relations between SF and GBSR, GAF and GGKS, and SGF and SLGF. The other depicted overlaps might be unsubstantiated.

of lower bounds for such succinctness gaps have one thing in common. In the used classes of particularly succinct sentences quantifier alternations played a key role that were not subject to the characteristic syntactic restrictions of the respective original fragment. For example, for SF versus BSR we used a class of SF sentences where the number of quantifier alternations was unbounded, whereas in BSR at most one quantifier alternation is allowed. In the SLGF-versus-LGF case, the considered class of SLGF sentences (which actually are SGF sentences) contains nested quantifiers of unbounded depth that do not adhere to the guardedness conditions imposed by LGF. Similarly, in the case SF versus the class of Gaifman-normal sentences, the class of sentences used for showing the non-elementary succinctness gap contains quantifier alternations of unbounded depth that are not allowed in first-order sentences in Gaifman normal form. Although the latter case does not fall into the category of extended decidable first-order fragment versus original decidable first-order fragment, it highlights a succinctness gap between a class of sentences whose definition is based on separateness compared to a certain class of first-order sentences with a restricted syntax. Although we have not investigated the succinctness gaps for all extended fragments, this does not mean that the unexamined gaps are only polynomial or smaller. For instance SGNFO versus GNFO seems to be a natural candidate for another non-elementary succinctness gap, like in the case of SLGF versus LGF. The case of SFL versus FL seems to be more tricky, though, as FL-Sat is

More succinct fragment	Less succinct fragment	Succinctness gap (lower bound)	Reference
SF	BSR	non-elementary	Theorem 3.2.7
SF	Gaifman-local first-order fragment	non-elementary	Theorem 3.3.18
GGKS	GKS	exponential	Theorem 3.9.9
SGF	LGF	non-elementary	Theorem 3.10.8
SFO^2	FO^2	exponential	Theorem 3.12.5

Table 12.1: Summary of the unconditional lower bounds regarding succinctness gaps that we have derived in the present thesis (cf. Table 1 on page 4).

computationally very hard and thus using FL alone one can already enforce very large domain sizes in a succinct manner. It remains to be investigated one can do significantly better using SFL syntax instead. For the case GAF versus AF we have only derived a bound conditioned on NExpTime $\neq$ ExpTime in Proposition 3.8.9, and an unconditional lower bound is missing for the time being. Similarly, the exponential lower bounds regarding the succinctness gaps between GGKS and GKS and between SFO^2 and FO^2 could possibly be improved. This is left for future work. One more aspect of the succinctness gaps that might be worth investigating in the future will be discussed in Section 12.1.2.

Our main method for proving decidability of the newly introduced first-order fragments is based on the mentioned equivalence-preserving translations into fragments that are already known to be decidable. In Chapter 4 we have complemented this syntactic point of view with a semantic perspective, based on an investigation of dependences between existentially and universally quantified variables in sentences. What we have found are weak dependences, which have a finite character. BSR, SF, and GBSR are special fragments in this respect, as all dependences in sentences from these fragments are weak. Conversely, we have observed in Theorem 4.2.1 that every sentence in which all dependences are weak has some equivalent in the BSR fragment; one may say that BSR semantically captures this class of sentences. Again, the BSR equivalents may be significantly longer than the initial sentence. The weakness of all occurring dependences leads to the property that any model of such a sentence, even if it has an infinite domain, has a finite substructure that is a model of the very same sentence. This highlights a difference in expressive power between BSR, SF, GBSR and other fragments, such as GAF: a GAF sentence can have a model without satisfying substructures, see Example 4.3.1. Hence, among the other applications that we have sketched previously, an analysis of weak dependences could perhaps also help discern expressive power when comparing two first-order fragments.

Speaking of applications for the analysis of weak dependences, Section 7.2 offered first insights concerning Skolemization techniques that are sensitive to weak dependences. Taking the Boolean structure of sentences into account in addition, e.g. in the spirit of Section 3.6, might lead to further improvements. As we have already pointed out in the beginning of that section, this might offer interesting and valuable directions of research automated reasoning could benefit from. Moreover, further investigations might yield new insight in proof complexity.

In addition to the already mentioned results, we have investigated the computational complexity of SF-Sat, GBSR-Sat, and some of their subproblems in detail. Figure 5.1 on page 141 depicts an overview and shows that both SF-Sat and GBSR-Sat have k-NExpTime-complete subfragments for every positive integer k. The respective unrestricted satisfiability problems are non-elementary, or, more precisely, Tower-complete (cf. Definition 5.0.2). Since the computational complexity of the satisfiability problem associated with the Horn and Krom subcases of the $\exists^*$ and $\exists^*\forall^*$ subfragments of SF and GBSR is significantly lower (unless some of the complexity classes NL, P, NP, PSpace, ExpTime, NExpTime coincide), we have formulated the conjecture that this behavior might continue on a larger scale — cf. Conjecture 5.2.2 and see also Figure 5.2 on page 149.

Furthermore, we have shown that BSR and AF are closed under Craig–Lyndon interpolation,

which immediately entails the same property for SF, GBSR, and GAF. From the research literature we have concluded that SGF and SLGF are not closed under interpolation, while the class of SGNFO sentences enjoys this property. This question is still unanswered for the other newly introduced first-order fragments.

Finally, we have sketched more ideas concerning applications of separateness in Chapter 7, ranging over topics such as investigations of the effects of separateness in interpreted logics, and the elimination of certain occurrences of second-order quantifiers.

In the following sections, we shall elaborate a bit on applications of some of the decidable fragments we have identified in the present thesis and we shall also sketch further ideas for future work.

12.1.1 Potential Applications for the Newly Introduced Decidable First-Order Fragments

In Part I of the present thesis we have concentrated on SF much more than on any other fragment. The reasons for this focus on SF are manifold: (a) it has been the first novel decidable first-order fragment discovered by the author, (b) its definition is simple and easy to handle technically, (c) it extends BSR, which found many applications, e.g. in verification, and has attracted quite some attention in the automated reasoning community.

The Bernays–Schönfinkel–Ramsey fragment has become popular in verification and in automated reasoning because it constitutes a good compromise between expressiveness and simplicity, in particular due to the finiteness of the Herbrand domain associated with any BSR sentence (after exhaustive Skolemization). The fragment is sometimes conceived as an intermediate syntactic step between propositional logic and full first-order logic. Compared to propositional logic, certain logical properties can be expressed exponentially more succinct in BSR. In the automated reasoning and verification communities the term *effectively propositional logic*, or *EPR* for short, has been put about as an alternative name for BSR.[1] There is, for instance, an EPR division at the annual *CADE ATP System Competition (CASC)* [SS06, Sut18]. The research literature offers a plethora of works based on BSR, extensions thereof, and variants of BSR in settings beyond first-order logic, e.g. [PV07a, PV07b, WPK09, CW10, EKKV10, EKK$^+$12, PO12, IBI$^+$13, PV13, BDMMS14, IBI$^+$14, IBR$^+$14, KBI$^+$15, PMP$^+$16, FMSZ17, RIS17]. In automated reasoning the finiteness of the Herbrand domain associated with BSR sentences is appreciated very much. There are dedicated reasoning approaches making use of this property in one form or another, see, e.g., [PV08, Hil08, GdM09, PdMB10, HW13, AW15]. Since these methods work well in practice, it is sometimes even worth to reduce more general first-order problems to BSR in order to apply finite model finders to the resulting formulas, see [BFdNT09], for instance.

All of the above said indicates that SF and its larger relative GBSR could be of great interest to the verification and automated reasoning communities, as both extend BSR and offer more syntactic freedom for modeling the behavior of systems and their properties. The formalizations that have been presented in Sections 3.3 and 5.3 are very instructive regarding what can be formalized in SF and GBSR and how.

In Section 10.3 we have discussed the combination of theories in the *Nelson–Oppen framework*. There is a series of papers by Fontaine and his collaborators [Fon07, Fon09, AF11, CFR14], where component theories are considered that are axiomatized using finite sets of sentences stemming exclusively from MFO$_\approx$, BSR, AF with equality, GF, LGF, or FO^2. The results by Fontaine et al. state that such theories are *gentle* ([Fon09], [AF11]), that is, for every set of literals $\mathcal{L}$ over $\mathcal{T}$'s vocabulary the *spectrum*[2] of $\mathcal{T} \cup \mathcal{L}$ can be computed and is either (a) a finite set of finite cardinalities or (b) the union of a finite set of finite cardinalities and all the (finite and infinite) cardinalities greater than a computable finite cardinality ([Fon09], Definition 3). One of the contributions

[1] Although the term EPR is used ambiguously throughout the literature, e.g. sometimes referring to BSR, sometimes only referring to BSR without equality, the alternative term should be kept in mind when looking up publications in which BSR is used in applications.

[2] The *spectrum* of a satisfiable sentence φ is the set of all cardinalities κ such that there is some model $\mathcal{A} \models \varphi$ whose domain A has cardinality κ.

in [Fon09] is that satisfiability with respect to two vocabulary-disjoint theories $\mathcal{T}_1, \mathcal{T}_2$ — that is, answering the question whether there is a model of $\mathcal{T}_1 \cup \mathcal{T}_2 \cup \{\varphi\}$ for any sentence $\varphi = \exists \bar{z}.\ \bigwedge_{i \in I} L_i$ over the combined vocabulary of $\mathcal{T}_1$ and $\mathcal{T}_2$ — is decidable in the following cases: $\mathcal{T}_1$ is gentle and (i) $\mathcal{T}_2$ is gentle as well, or (ii) $\mathcal{T}_2$ is a finitely axiomatized first-order theory, or (iii) $\mathcal{T}_2$ is a decidable theory that only admits a fixed finite (possibly empty) known set of finite cardinalities for its models, and possibly infinite models ([Fon09], Theorem 3). According to Fontaine, examples for such theories $\mathcal{T}_2$ are real or integer linear arithmetic and certain known theories over array data structures. In [CFR14], some of the mentioned results are extended to a setting with theories that are not built over disjoint vocabularies but may share unary predicate symbols.

Since we have shown that (a) SF and GBSR are equivalent to BSR, (b) GAF with equality is equivalent to AF with equality, (c) SGF is equivalent to GF, (d) SLGF is equivalent to LGF, and (e) SFO^2 is equivalent to FO^2, the combination results obtained by Fontaine et al. are also applicable to theories that are axiomatized using finite sets of sentences exclusively taken from SF, GBSR, GAF, SGF, SLGF, or SFO^2, respectively.

12.1.2 More about Future Work

There are some obvious omissions in the material covered in the present thesis that are worth being filled in in future investigations. Examples include (a) pinning down the computational complexity of satisfiability for the rest of the introduced decidable fragments, (b) bounding the blowup incurred when translating sentences from the extended fragments to the underlying original fragments, (c) investigating all the new fragments apart from SF, GBSR, and GAF under the semantic lens and taking weakness of dependences into account, e.g. the interplay of weak dependences with guardedness, (d) checking these fragments for closedness under interpolation and checking whether BSR, SF, GBSR, AF, GAF are still closed under interpolation in the presence of equality, and (e) using separateness to extend further decidable first-order fragments such as Maslov's K. Moreover, separateness may turn out to be even more versatile in future investigations. We have already discussed three possible directions in Chapter 7: the effects of separateness in interpreted logics, Skolemization techniques enhanced by a certain sensitivity to weak dependences, and the elimination of certain occurrences of second-order quantifiers. Other topics that we have touched only very briefly and that might be worth further investigation are the interplay between Boolean structure and separateness, see Section 3.6, and the possible connections of weak dependences to the field of dependence logic (broadly construed), including independence-friendly logic, logics with Henkin quantifiers, and other related research fields, see Remark 7.2.3 on page 192 for references. It is also worth pointing out that for most of the decidable fragments we have extended in Chapter 3 there are resolution-based decision procedures known (consult Chapter 3, pages 23–28, for references). It would be interesting to know whether and how existing procedures could be adapted so as to cope with separateness and become decision procedures for the extended fragments as well.

To conclude the present section, we shall sketch one more idea. We have emphasized time and again that, compared to BSR sentences, SF sentences can express certain logical properties much more succinctly. This holds true in particular for properties that exhibit a high degree of *structural regularity*. An example for such a property is the one described by the family of SF sentences $\big(\varphi_n\big)_{n \geq 1}$ with

$$\varphi_n := \forall x_n \exists y_n \ldots \forall x_1 \exists y_1.\ \textstyle\bigwedge_{i=1}^{n} \big(P_i(x_1, \ldots, x_n) \leftrightarrow Q_i(y_1, \ldots, y_n)\big).$$

We have already encountered a variant of this class of sentences in the proof of Theorem 3.2.7. Although the domain size of the following family of models $\big(\mathcal{A}_n\big)_{n \geq 1}$ with $\mathcal{A}_n \models \varphi_n$ for every n grows massively with increasing n, its interpretation of the predicate symbols P_i and Q_i is given by a rather simple pattern and, hence, each $\mathcal{A}_n$ is intuitively very regular — the latter is witnessed by the shortness of the following definition of $\mathcal{A}_n$:

$$\mathsf{A}_n := \textstyle\bigcup_{k=1}^{n} \big\{ \mathsf{a}_S^{(k)}, \mathsf{b}_S^{(k)} \;\big|\; S \in \mathcal{P}^k[n] \big\},$$

$$P_i^{\mathcal{A}_n} := \big\{ \langle \mathsf{a}_{S_1}^{(1)}, \ldots, \mathsf{a}_{S_n}^{(n)} \rangle \in \mathsf{A}^n \;\big|\; i \in S_1 \in S_2 \in \ldots \in S_n \big\} \text{ for } i = 1, \ldots, n, \text{ and}$$

$Q_i^{\mathcal{A}_n} := \big\{ \langle \mathsf{b}_{S_1}^{(1)}, \ldots, \mathsf{b}_{S_n}^{(n)} \rangle \in \mathsf{A}^n \;\big|\; i \in S_1 \in S_2 \in \ldots \in S_n \big\}$ for $i = 1, \ldots, n$.

Any of the structures $\mathcal{A}_n$ neatly captures the essence of the logical property described by φ_n, as every domain element $\mathsf{a}_S^{(k)}$ has a corresponding twin element $\mathsf{b}_S^{(k)}$ that mirrors in the predicates $Q_i^{\mathcal{A}_n}$ exactly the role that $\mathsf{a}_S^{(k)}$ plays in the predicates $P_i^{\mathcal{A}_n}$.

More generally, consider any logical property π_n that is parameterized by some positive integer n and that can be expressed by a (uniform) family of BSR sentences. Let $f(n)$ be the function representing the length of a shortest BSR sentence ψ that describes π_n. Let $g(n)$ be the function that denotes the length of a shortest SF sentence describing π_n. We know that there are properties π_n such that $g(n)$ can be bounded from above by some polynomial but we cannot find any integer k such that $f(n)$ is bounded from above by some k-fold exponential function. In such a case we would say that π_n is structurally fairly regular, as we can describe it with an SF sentence of polynomial length. Now imagine a property π'_n accompanied with corresponding functions $f'(n)$ and $g'(n)$ for which we have $g'(n) \in \Omega\big(f'(n)\big)$, i.e. the length of shortest SF sentences describing π'_n is asymptotically of the same order as the length of shortest BSR sentences describing π'_n. On an intuitive level, this means that the relaxed syntactic conditions of SF do not provide a significant edge over BSR when π'_n is to be described. For instance, the possibility to use quantifier alternations within the limits of SF does not help to formulate an asymptotically shorter description of π'_n. It seems that π'_n requires a more sophisticated and lengthy description than, for instance, π_n does, or, viewed from the opposite angle, π'_n exhibits a lower degree of structural regularity than π_n. A possible measure for this lack of regularity might be provided by the gap between $f'(n)$ and $g'(n)$: the smaller the gap, the higher the *structural irregularity* of π'_n.

Instead of the comparison SF versus BSR, one could also use the comparison between SF sentences and equivalent Gaifman-local sentences. Of course, the above said is also relevant to other fragments and not exclusively to SF, for instance, to SLGF versus LGF or to the full class of relational first-order sentences versus relational Gaifman-local sentences. We have already encountered a number of examples of structurally fairly regular properties described by SF sentences in the preceding chapters, e.g. in Section 3.2 in the proof of Theorem 3.2.7 (the property described by the sentence φ), during the preparations for the proofs of Theorems 3.3.11 and 3.3.18 in Section 3.3.3 (the property described by the $\chi_{m,k}$ on page 43) and in Chapter 5, Section 5.3.1 (the property described by the sentence $\psi_1 \wedge \ldots \wedge \psi_{16}$, for instance), which contains the heart of the proof of Theorem 5.3.11.

The general idea of measuring structural regularity by means of the asymptotic length of shortest logical descriptions appears to have some similarity to concepts investigated in the field of algorithmic information theory and Kolmogorov complexity in particular (see, e.g., the textbooks [DH10] and [Cal02] for introductory material). Potential connections and interrelations remain to be studied.

12.2 First-Order Linear Arithmetic with Uninterpreted Predicates

In Part II of the present thesis we have explored the decidability boundary for first-order linear arithmetic with uninterpreted predicate symbols. On the decidable side (Chapter 10) we have mainly focused on the domain of the rational numbers and have introduced two fragments of the language for which the satisfiability problem is decidable: *BSR with simple linear rational constraints (BSR(SLR))* and *BSR with bounded difference constraints (BSR(BD))*. The two can be conceived as extensions of the Bernays–Schönfinkel–Ramsey fragment enhanced with certain linear rational arithmetic expressions. Indeed, we have shown that checking satisfiability is NExpTime-complete for both fragments. The proof strategy is very similar for both cases. Although a finite model property in the usual sense cannot be established due to the inherent infiniteness of the underlying domain, we have derived a property with a similar flavor. In a first step, we have identified equivalence relations $\sim$ over $\mathbb{Q}^m$ that induce only finitely many equivalence classes, each containing m-tuples that are pairwise indistinguishable from the perspective of the the admitted

arithmetic atoms. Then, we have proved that it is sufficient to consider only candidate models $\mathcal{A}$ that are *uniform* in the sense that the interpretation of predicate symbols, e.g. $P : \mathbb{Q}^m$, does not distinguish $\sim$-equivalent tuples either: for any two such tuples $\bar{r}_1, \bar{r}_2$ we have $\bar{r}_1 \in P^{\mathcal{A}}$ if and only if $\bar{r}_2 \in P^{\mathcal{A}}$. For every satisfiable finite clause set over the language of BSR(SLR) or BSR(BD) there is such a uniform model, and, moreover, this model can be described my finite means. Based on this observation, we have devised computable transformations from finite BSR(SLR) and BSR(BD) clause sets into equisatisfiable finite BSR clause sets without interpreted symbols, except for equality.

On the negative side (Chapter 11) of the decidability boundary, we have identified several fragments with a satisfiability problem that is undecidable or, in some cases, not even semi-decidable. We have treated settings over different arithmetic domains: linear arithmetic over the natural numbers, the rationals, and the reals. Moreover, in many cases it has turned out that a single uninterpreted predicate symbol of arity one suffices to encode the halting problem for two-counter machines on given inputs. We have tried to keep the number of quantifier alternations and quantifiers at a minimum. An overview of the most important results is given in Table 12.2. In Sections 10.4 and 11.4, we have studied the decidability boundary around one particular

Fragment description	Result	References
$\forall^*$-PA+P and $\forall^*$-LRA+P	undec., but Unsat is r.e.	Theorems 11.2.2, 11.2.6, 11.3.3
$\forall$-Horn–Krom PA+P	undecidable	Theorem 11.2.4
$\forall^2\exists$-PA+P and $\forall^2\exists$-LRA+P	undec. and not r.e.	Theorems 11.3.7, 11.3.8
$\forall\exists$-PA+P	undec. and not r.e.	Theorem 11.3.9
$\exists^2\forall^*$-LRA+PN over $[0,1]$, difference constraints only, N, P interpreted with finite sets	undec., but Sat is r.e.	Theorem 11.4.4

Table 12.2: Summary of the most important undecidability results obtained in Chapter 11. PA+P stands for *Presburger arithmetic with an uninterpreted unary predicate symbol* P. LRA+PN abbreviates *linear rational arithmetic with two uninterpreted unary predicate symbols* P, N; LRA+P stands for the restriction of the latter to only one such predicate symbol. The fragment in the last line is a restricted form of LRA+PN where (a) the domain is restricted to the rational unit interval $[0,1]$, (b) all arithmetic atoms have the form $x_0 = 0$ or $x - y \triangleleft c$ where x, y are universally quantified variables, c is either a rational number or an uninterpreted constant symbol and $\triangleleft$ ranges over the relations $<, \leq, =, \neq, \geq, >$, and (c) the interpretation of the predicate symbols P, N is restricted to finite subsets of $[0,1]$. The abbreviation r.e. stands for *recursively enumerable*, as usual, a synonym for semi-decidable. The terms *Unsat* and *Sat* address the set of unsatisfiable sentences and the set of satisfiable sentences from the respective fragment.

kind of arithmetic atoms very closely, namely around *difference constraints*, that is, atoms of the form $x - y \triangleleft c$ with universally quantified variables x, y, some integer c, and any relation $\triangleleft \in \{<, \leq, =, \neq, \geq, >\}$. On the one hand, we have shown decidability of BSR(BD), where every atom $x - y \triangleleft c$ needs to be conjoined with bounds $c_x \leq x \wedge x \leq d_x \wedge c_y \leq y \wedge y \leq d_y$ regarding the range of x and y. On the other hand, we have shown that this fragment becomes undecidable as soon as we either drop the bounds on x and y, or as soon as we allow c to be an uninterpreted constant symbol or an existentially quantified variable.

For some of the undecidable fragments we have been able to show that satisfiability and unsatisfiability are not even semi-decidable (cf. Table 12.2). To this end, we have encoded the *recurrence problem* for two-counter machines, which required a $\forall\exists$ quantifier alternation. Such a high degree of undecidability, has immediate consequences for automated reasoning. Whenever decision procedures cannot be constructed, then one could still hope for a semi-decision procedure in the form of a *sound* deductive calculus that is either *complete* — every logical consequence is derivable — or *refutationally complete* — *logical falsity* is derivable from any inconsistent set

of formulas. However, if a satisfiability problem or an unsatisfiability problem is even not semi-decidable, then such calculi cannot exist — they have to be unsound or incomplete or even both. In this situation, the best one could hope for is sound heuristics that perform reasonably well on certain problem instances.

Apart from their theoretical value, such negative results are relevant for several areas of verification where variants and extensions of first-order arithmetic with uninterpreted function or predicate symbols play a role. In Section 11.5 we have elaborated on the implications for the Bernays–Schönfinkel fragment of separation logic, quantified theories of data structures, arrays in particular, and quantified combinations of the theory of *equality over uninterpreted functions* with fragments of Presburger arithmetic. Moreover, we have argued that in certain settings we cannot even hope for refutationally complete deductive calculi. In such cases we either have to content ourselves with heuristics instead of sound and complete reasoning methods or formulate restricted fragments having less hard (un)satisfiability problems.

12.2.1 Applications for the New Decidable Fragments and Future Work

We have already mentioned that BSR has found many applications, e.g. in the field of verification of hardware and software (cf. Section 12.1.1). Moreover, we have outlined applications for various fragments of first-order arithmetic with or without uninterpreted predicate and function symbols, see Chapter 8, in particular Remarks 8.0.1 and 8.0.2 and the part on related work at the end of the section; see also the beginning of Section 10.5, and Section 11.5. The application areas we have encountered so far include scheduling problems, program analysis, and modeling and verification of data structures and timed systems. In the light of this success, it seems likely that BSR(SLR) and BSR(BD) could turn out to be useful in a broad variety of applications as well. Since difference constraints have been of use in the analysis and verification of timed systems (cf. Remark 8.0.2), the idea suggests itself that BSR(BD) may find applications in this area. Indeed, we have shown in Section 10.5 that reachability for timed automata can be expressed with BSR(BD), although not entirely in a straightforward fashion. To this end, we have slightly relaxed the usual notion of synchronous progression of all clocks. Our modifications do not affect the reachability relation. It is to be expected that BSR(BD) lends itself to even more sophisticated applications in the area of timed systems or other fields. A further potential area of application for BSR(SLR) or BSR(BD) is the representation of temporal precedence in ontologies and, more general, temporal reasoning in knowledge representation. For instance, in [SWW10, Wis12] the authors have demonstrated that a core of the large ontology named YAGO [SKW08, HSBW13, RSH$^+$16] can be translated into a subfragment of BSR in a semantic-preserving way. This fragment was chosen, since reasoning procedures are available that work sufficiently well in practice. However, the authors also made clear that temporal information had to be disregarded at that time. Clearly, BSR(SLR) offers ways to encode temporal precedence and invites reasoning about temporal knowledge, if suitable calculi were to be developed and implemented, e.g. based on superposition modulo (linear) rational arithmetic [AKW09, EKK$^+$11, Kru13]. Very first steps have been proposed in [KW12], Section 5.

In Section 10.3 we have slightly shifted our perception of BSR(SLR) and looked at it from the perspective of the Nelson–Oppen combination framework. The setting then presented itself as a combination of existential linear rational arithmetic with the BSR theory enhanced with a dense linear ordering — notice that density is not finitely axiomatizable in BSR. As the interpreted predicate symbols $<, \leq$ are shared by the two constituent theories, our setting in fact lies beyond the scope of the Nelson–Oppen framework. Hence, the results we obtained in Section 10.3 constitute a contribution to the field of non-signature-disjoint combination frameworks. This particular point of view made it easy to describe extensions of BSR(SLR) for which satisfiability is still decidable. One such example is GBSR(SLR), which is based on the generalization of BSR presented in Chapter 3 (cf. Definition 10.3.3 and Corollary 10.3.4). Another example is the fragment described in Theorem 10.3.2. By Proposition 3.4.4, this immediately entails that also a combination with SF or MFO yields decidable satisfiability problems. This is one example showing that separateness of first-order variables also facilitates decidable extensions of decidable fragments in interpreted settings. It turns out that also the arithmetic side can be extended. For example, universal

quantification can be allowed also on the arithmetic side under certain circumstances. Furthermore, one could consider polynomials over the real numbers (cf. Theorem 10.3.2), where quantifier-elimination procedures are available. It seems likely that the combination-of-theories point of view has even more potential, which might be worth exploring further.

Although the case of BSR(BD) is different in the sense that it cannot be re-formulated as a combination of theories in an obvious way, it should be possible to extend it to a GBSR variant as well. On the other hand, it seems to be less clear how to extend the arithmetic side of BSR(BD) significantly so as to get a more expressive decidable fragment. This remains to be investigated in future work. However, this may require more advanced proof techniques. Our approach in Chapter 10 is based on the fact that it is sufficient to consider structures that are uniform with respect to a suitable equivalence relation $\sim$ that induces only finitely many equivalence classes. It might be necessary to go beyond uniformity. For example, one might consider *"ultimately periodic"* structures instead of ones that are uniform — an appropriate definition of the former should subsume the latter as a special case, and, more importantly, it should allow a description of the structure by finite means. A set $S \subseteq \mathbb{Z}$ is called *ultimately periodic with period* p if there is some $t \in \mathbb{N}$ such that for every $r \geq t$ we have $r \in S$ if and only if $r + p \in S$, and for every $r \leq -t$ we have $r \in S$ if and only if $r - p \in S$. Such sets capture the expressiveness of Presburger arithmetic: A set of natural numbers is definable in Presburger arithmetic if and only if it is ultimately periodic ([End72], Theorem 32F). In the realm of the rational or real numbers one may have to add a second parameter $g \in \mathbb{N}_{\geq 1}$, for *granularity*, alongside the period and require some uniformity property similar to the following. Every interval $\left(q + \frac{d}{g}, q + \frac{d+1}{g}\right)$ with $q, d \in \mathbb{Z}$ and $0 \leq d \leq g - 1$ either entirely belongs to the periodic set or it is disjoint from the set. No matter how the definition is to be formulated in detail, the key property will be that any sets satisfying the property can be described by finite means with a computable bound regarding the length of the description, just like the uniform structures we have been using for BSR(SLR) and BSR(BD).

Another possible direction for extending BSR(SLR) or BSR(BD) is the addition of uninterpreted function symbols. Steps in this direction have been made, e.g. in [GdM09, HVW17a] over the integer domain and essentially for *stratified vocabularies* (cf. Section 3.14.2). Both approaches yield extensions of the *array property fragment* (cf. Remark 8.0.1). From that fragment it is known that arithmetic atoms need to be more restricted than in BSR(SLR). For instance, atoms of the form $\neg x < y$ with universally quantified integer variables x, y are not admitted in clauses but $\neg x \leq y$ is. Not adhering to these restrictions yields an undecidable satisfiability problem, cf. Theorem 2.4.2 in [Bra07] and Theorem 11.16 in [BM07].

Finally, there are plenty of decidable first-order fragments besides BSR and its separated extensions SF and GBSR that might serve as a basis for decidable fragments of first-order arithmetic with uninterpreted predicate or function symbols, see Chapter 3. There is quite some research to be done in this direction.

12.2.2 Automated Reasoning in Practice: Instantiation Methods for BSR(SLR) and BSR(BD)

As the analysis of the computational complexity of decision problems mostly focuses on worst-case scenarios, it hardly comes as a surprise that solving problem instances originating from practical applications does not necessarily need as much time or space as the theoretical worst-case analysis would predict. It is meanwhile a broadly accepted fact that automated reasoning in propositional logic, quantified Boolean logic, the Bernays–Schönfinkel fragment, or combinations of theories, to name a few prominent examples, can be feasible in practice. This is in spite of the fact that the traditional narrative of complexity theory claims that problems beyond the NP-hardness barrier ought to be considered infeasible. Thanks to a great engineering effort over the last decades, we have potent methodologies available today to make automated reasoning work in practice, see [RV01, Bie09, CHVB18, HS18]

Instantiation of universally quantified variables is one technique that is being used in automated reasoning tools [KS10, Kor13a, RKK17, Bar17, RBF18], e.g. for reasoning in the Bernays–Schönfinkel–Ramsey fragment or linear arithmetic. Moreover, instantiation is the method of

choice to decide satisfiability for the array property fragment in [BMS06, Bra07, GdM09], for instance. Independently from these developments, but yet along the same lines, the author of the present thesis and two co-authors have devised improved instantiation methods for close relatives of BSR(SLR) [VW15, HVW17a], which is, in turn, related to the array property fragment. The approach should be transferable to BSR(SLR) and BSR(BD) but will most likely get slightly more complicated. We shall outline the key ideas below. A full presentation of the results lies beyond the scope of the present thesis.

We shall concentrate on subfragments of BSR(SLR) over the rational and the integer domain where the main syntactic restriction is the following. In addition to the restrictions imposed by the definition of BSR(SLR) (Definition 10.0.1), we restrict the options for the predicate symbol $\triangleleft$ in arithmetic atoms of the form $x \triangleleft y$ to $\leq$, $=$, or $\geq$, whenever x, y are universally quantified variables of sort $\mathbb{Q}$ or $\mathbb{Z}$. The rationale behind this restriction is twofold: simplicity and (practical) efficiency of instantiation. The key difference is that we do not have to handle $\sim$-equivalent tuples of rationals but it is sufficient to consider individual rational numbers — one-tuples so to say. In the two-dimensional case illustrated in Figure 12.2, this means that we can safely ignore the triangles that emerge around the diagonal and rather only consider a division of the rational plane into bounded and unbounded rectangular regions. It is to be expected that there are also good instantiation techniques for BSR(SLR) without additional syntax restrictions and also for BSR(BD). This direction of research is left for future work.

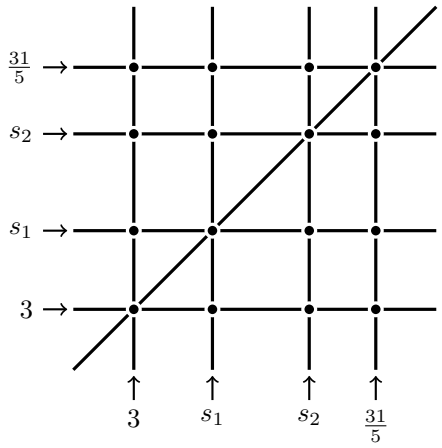


Figure 12.2: Partition of the two-dimensional rational plane into $\sim_1$-equivalence classes (cf. Chapter 8, page 221) with respect to two unspecified rational values s_1, s_2 lying between 3 and $\frac{31}{5}$. Every dot, line segment, rectangular white area, and triangular white area represents an equivalence class induced by $\sim_1$.

The improved instantiation methods presented in [VW15, HVW17a] are based on (i) a detailed analysis of which arguments of predicate symbols are affected by which arithmetic constraints, (ii) optimizations inspired by well-established quantifier-elimination techniques which concern the kind of constraints that need to be taken into account, and (iii) the observation that we can apply different optimizations for sufficiently disconnected argument positions. All in all, one can significantly reduce the number of instances that need to be generated to decide satisfiability, compared to the number of instances less sophisticated instantiation methods produce that are used to decide satisfiability for similar logic fragments, see, for example, [BMS06, Bra07, GdM09].

Example 12.2.1. *Consider the sentence*

$$\varphi := \exists z \forall u x_1 x_2 y_1 y_2. \big(x_2 \neq 5 \wedge R(x_1) \rightarrow Q(u, x_2)\big) \wedge \big(y_1 < 7 \wedge y_2 \leq 2 \rightarrow Q(z, y_2) \vee R(y_1)\big)$$

where the variables z and u are of an uninterpreted sort and the x_i, y_i are of sort $\mathbb{Z}$. The results presented in [HVW17a] reveal that this sentence is satisfiable over the integers if and only if the

following sentence is satisfiable over the integers:

$$\begin{array}{rl} \varphi' := \exists z. & \Big(\quad 5+1 \neq 5 \wedge R(c_{-\infty}) \;\rightarrow\; Q(z, 5+1)\Big) \\ \wedge & \Big(\quad c_{-\infty} \neq 5 \wedge R(c_{-\infty}) \;\rightarrow\; Q(z, c_{-\infty})\Big) \\ \wedge & \Big(c_{-\infty} < 7 \wedge 5+1 \leq 2 \;\rightarrow\; Q(z, 5+1) \vee R(c_{-\infty})\Big) \\ \wedge & \Big(c_{-\infty} < 7 \wedge c_{-\infty} \leq 2 \;\rightarrow\; Q(z, c_{-\infty}) \vee R(c_{-\infty})\Big) \\ \wedge & \qquad\qquad c_{-\infty} < 2 \;. \end{array}$$

The sentence φ' has been derived from φ by the following instantiation steps: (1) u has been instantiated with the existentially quantified variable z, (2) x_2 and y_2 have been instantiated with the (abstract) integer values $5+1$ and $-\infty$, and (3) x_1 and y_1 have been instantiated with $-\infty$ only. The instantiation does not need to consider any instantiation point derived from upper bounds $y_1 < 7$, $y_2 \leq 2$, because it is sufficient to explore the integers either from $-\infty$ upwards — in this case upper *bounds on integer variables can be ignored — or from $+\infty$ downwards — ignoring* lower *bounds.*[3] *Moreover, instantiation does not need to consider the value $5+1$ for x_1 and y_1, motivated by the fact that in the first conjunct of φ the argument x_1 of R is not affected by the constraint $x_2 \neq 5$. The abstract values $-\infty$ and $+\infty$ are represented by fresh uninterpreted constant symbols, together with defining axioms. For the example, we introduce the fresh Skolem constant $c_{-\infty}$ to represent $-\infty$ (a "sufficiently small" value) together with the axiom $c_{-\infty} < 2$, where 2 is the smallest integer occurring in φ.*

If we consider φ over the rational domain, instantiation leads to a sentence φ'' that is slightly different from φ':

$$\begin{array}{rl} \varphi' := \exists z. & \Big(\quad c_{5+\varepsilon} \neq 5 \wedge R(c_{-\infty}) \;\rightarrow\; Q(z, c_{5+\varepsilon})\Big) \\ \wedge & \Big(\quad c_{-\infty} \neq 5 \wedge R(c_{-\infty}) \;\rightarrow\; Q(z, c_{-\infty})\Big) \\ \wedge & \Big(c_{-\infty} < 7 \wedge c_{5+\varepsilon} \leq 2 \;\rightarrow\; Q(z, c_{5+\varepsilon}) \vee R(c_{-\infty})\Big) \\ \wedge & \Big(c_{-\infty} < 7 \wedge c_{-\infty} \leq 2 \;\rightarrow\; Q(z, c_{-\infty}) \vee R(c_{-\infty})\Big) \\ \wedge & \qquad\qquad c_{-\infty} < 2 \\ \wedge & \qquad 5 < c_{5+\varepsilon} \wedge c_{5+\varepsilon} < 7 \;. \end{array}$$

The difference between φ'' and φ' is that the expressions $5+1$ have been replaced with the constant symbol $c_{5+\varepsilon}$. This constant symbol is intended to represent some value that is "just a little larger" than 5. That is, the value of $c_{5+\varepsilon}$ is supposed to be larger than 5 but smaller than all occurring rational values larger than 5. This is why the axiom $5 < c_{5+\varepsilon} \wedge c_{5+\varepsilon} < 7$ needs to be added to φ''.

In [HVW17a] it is shown in addition that the outlined instantiation methods are compatible with uninterpreted function symbols and additional background theories under certain syntactic restrictions. These results are based on an (un)satifiability-preserving embedding of uninterpreted function symbols into BSR. There are interesting known logic fragments that fall into this syntactic category: many-sorted first-order sentences over *stratified vocabularies* [ARS07, ARS10, Kor13b] — see also Section 3.14.2 —, the *array property fragment* [BMS06, Bra07], and the *finite essentially uninterpreted fragment*, possibly extended with simple integer arithmetic [GdM09]. Consequently, reasoning procedures for these fragments that employ forms of instantiation may benefit from the outlined instantiation approach.

Regarding automated reasoning techniques that work well in practice, it might be interesting to devise decision procedures for BSR(SLR), BSR(BD), or other fragments of first-order arithmetic with uninterpreted function or predicate symbols based on hierarchic superposition coupled with strong instantiation methods. Starting points for such an endeavor can be found in [KW12, FW12, Kru13, Fie13].

[3] This trick is inspired by optimizations used in the field of linear quantifier elimination over the reals [LW93].

Bibliography

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity – A Modern Approach*. Cambridge University Press, 2009.

[ABRS09] Alessandro Armando, Maria Paola Bonacina, Silvio Ranise, and Stephan Schulz. New Results on Rewrite-Based Satisfiability Procedures. *ACM Transactions on Computational Logic*, 10(1), 2009.

[ACGM04] Alessandro Armando, Claudio Castellini, Enrico Giunchiglia, and Marco Maratea. A SAT-based Decision Procedure for the Boolean Combination of Difference Constraints. In *Theory and Applications of Satisfiability Testing (SAT'04), Revised Selected Papers*, 2004.

[Ack28] Wilhelm Ackermann. Über die Erfüllbarkeit gewisser Zählausdrücke. *Mathematische Annalen*, 100:638–649, 1928.

[Ack35] Wilhelm Ackermann. Untersuchungen über das Eliminationsproblem der mathematischen Logik. *Mathematische Annalen*, 110:390–413, 1935.

[Ack54] Wilhelm Ackermann. *Solvable Cases of the Decision Problem*. North-Holland, 1954.

[AD90] Rajeev Alur and David L. Dill. Automata For Modeling Real-Time Systems. In *Automata, Languages and Programming (ICALP'90)*, pages 322–335, 1990.

[AD94] Rajeev Alur and David L. Dill. A Theory of Timed Automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[AF11] Carlos Areces and Pascal Fontaine. Combining Theories: The Ackerman and Guarded Fragments. In *Frontiers of Combining Systems (FroCoS'11)*, pages 40–54, 2011.

[AG74] Stål Aanderaa and Warren D. Goldfarb. The Finite Controllability of the Maslov Case. *Journal of Symbolic Logic*, 39(3):509–518, 1974.

[AH94] Rajeev Alur and Thomas A. Henzinger. A Really Temporal Logic. *Journal of the ACM*, 41(1):181–204, 1994.

[AKVV16] Samson Abramsky, Juha Kontinen, Jouko Väänänen, and Heribert Vollmer, editors. *Dependence Logic, Theory and Applications*. Springer, 2016.

[AKW09] Ernst Althaus, Evgeny Kruglov, and Christoph Weidenbach. Superposition Modulo Linear Arithmetic SUP(LA). In *Frontiers of Combining Systems (FroCoS'09)*, pages 84–99, 2009.

[ALM17] Giovanni Amendola, Nicola Leone, and Marco Manna. Finite model reasoning over existential rules. *Theory and Practice of Logic Programming*, 17(5-6):726–743, 2017.

[ANvB98] Hajnal Andréka, István Németi, and Johan van Benthem. Modal Languages and Bounded Fragments of Predicate Logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.

[ARS07] Aharon Abadi, Alexander Moshe Rabinovich, and Mooly Sagiv. Decidable Fragments of Many-Sorted Logic. In *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'07)*, pages 17–31, 2007.

[ARS10] Aharon Abadi, Alexander Rabinovich, and Mooly Sagiv. Decidable Fragments of Many-Sorted Logic. *Journal of Symbolic Computation*, 45(2):153–172, 2010.

[AW15] Gábor Alagi and Christoph Weidenbach. NRCL – A Model Building Approach to the Bernays–Schönfinkel Fragment. In *Frontiers of Combining Systems (FroCoS'15)*, LNCS 9322, pages 69–84. Springer, 2015.

[Bar17] Haniel Barbosa. *New techniques for instantiation and proof production in SMT solving. (Nouvelles techniques pour l'instanciation et la production des preuves dans SMT).* PhD thesis, University of Lorraine, Nancy, France, 2017.

[BB16] Olaf Beyersdorff and Joshua Blinkhorn. Dependency Schemes in QBF Calculi: Semantics and Soundness. In *Principles and Practice of Constraint Programming (CP'16)*, pages 96–112, 2016.

[BBJ02] George S. Boolos, John P. Burgess, and Richard C. Jeffrey. *Computability and Logic.* Cambridge University Press, fourth edition, 2002.

[BBMR15] Jean-François Baget, Meghyn Bienvenu, Marie-Laure Mugnier, and Swan Rocher. Combining Existential Rules and Transitivity: Next Steps. In *Artificial Intelligence (IJCAI'15)*, pages 2720–2726, 2015.

[BBtC13] Vince Bárány, Michael Benedikt, and Balder ten Cate. Rewriting Guarded Negation Queries. In *Mathematical Foundations of Computer Science (MFCS'13)*, pages 98–110, 2013.

[BDMMS14] Davide Bresolin, Dario Della Monica, Angelo Montanari, and Guido Sciavicco. The light side of interval temporal logic: the Bernays–Schönfinkel fragment of CDT. *Annals of Mathematics and Artificial Intelligence*, 71(1-3):11–39, 2014.

[BdRV02] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, 2002.

[Beh22] Heinrich Behmann. Beiträge zur Algebra der Logik, insbesondere zum Entscheidungsproblem. *Mathematische Annalen*, 86(3–4):163–229, 1922.

[BEL01] Matthias Baaz, Uwe Egly, and Alexander Leitsch. Normal Form Transformations. In Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, volume I, pages 273–333. Elsevier and MIT Press, 2001.

[Ber66] Robert Berger. The undecidability of the domino problem. *Memoirs of the American Mathematical Society*, (66), 1966.

[Ber80] Leonard Berman. The Complexitiy of Logical Theories. *Theoretical Computer Science*, 11:71–77, 1980.

[BFdNT09] Peter Baumgartner, Alexander Fuchs, Hans de Nivelle, and Cesare Tinelli. Computing finite models by reduction to function-free clause logic. *Journal of Applied Logic*, 7(1):58–74, 2009.

[BFL94] Matthias Baaz, Christian G. Fermüller, and Alexander Leitsch. A Non-Elementary Speed-Up in Proof Length by Structural Clause Form Transformation. In *Logic in Computer Science (LICS'94)*, pages 213–219, 1994.

[BFL+18] Patricia Bouyer, Uli Fahrenberg, Kim Guldstrand Larsen, Nicolas Markey, Joël Ouaknine, and James Worrell. Model Checking Real-Time Systems. In Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking*, pages 1001–1046. Springer, 2018.

[BG01] Leo Bachmair and Harald Ganzinger. Resolution Theorem Proving. In Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, volume I, pages 19–99. Elsevier and MIT Press, 2001.

[BGG97] Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer, 1997.

[BGMR14] Jean-François Baget, Fabien Garreau, Marie-Laure Mugnier, and Swan Rocher. Extending Acyclicity Notions for Existential Rules. In *European Conference on Artificial Intelligence (ECAI'14)*, pages 39–44, 2014.

[BGW92] Leo Bachmair, Harald Ganzinger, and Uwe Waldmann. Theorem Proving for Hierarchic First-Order Theories. In *Algebraic and Logic Programming (ALP'92)*, pages 420–434, 1992.

[BGW93] Leo Bachmair, Harald Ganzinger, and Uwe Waldmann. Superposition with Simplification as a Desision Procedure for the Monadic Class with Equality. In *Computational Logic and Proof Theory, Third Kurt Gödel Colloquium (KGC'93)*, pages 83–96, 1993.

[BGW94] Leo Bachmair, Harald Ganzinger, and Uwe Waldmann. Refutational Theorem Proving for Hierarchic First-Order Theories. *Applicable Algebra in Engineering, Communication and Computing*, 5:193–212, 1994.

[Bie09] Armin Biere. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.

[BJ15a] Maria Paola Bonacina and Moa Johansson. Interpolation Systems for Ground Proofs in Automated Deduction: a Survey. *Journal of Automated Reasoning*, 54(4):353–390, 2015.

[BJ15b] Maria Paola Bonacina and Moa Johansson. On Interpolation in Automated Theorem Proving. *Journal of Automated Reasoning*, 54(1):69–97, 2015.

[BK08] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008.

[BL94] Matthias Baaz and Alexander Leitsch. On Skolemization and Proof Complexity. *Fundamenta Informaticae*, 20(4):353–379, 1994.

[BL11] Matthias Baaz and Alexander Leitsch. *Methods of Cut-Elimination*, volume 34 of *Trends in Logic*. Springer, 2011.

[BLM10] Jean-François Baget, Michel Leclère, and Marie-Laure Mugnier. Walking the Decidability Line for Rules with Existential Variables. In *Knowledge Representation and Reasoning (KR'10)*, 2010.

[BLM+17] Patricia Bouyer, François Laroussinie, Nicolas Markey, Joël Ouaknine, and James Worrell. Timed Temporal Logics. In *Models, Algorithms, Logics and Tools – Essays Dedicated to Kim Guldstrand Larsen on the Occasion of His 60th Birthday*, pages 211–230, 2017.

[BLS02] Randal E. Bryant, Shuvendu K. Lahiri, and Sanjit A. Seshia. Modeling and Verifying Systems Using a Logic of Counter Arithmetic with Lambda Expressions and Uninterpreted Functions. In *Computer Aided Verification (CAV'02)*, pages 78–92, 2002.

[BM07] Aaron R. Bradley and Zohar Manna. *The Calculus of Computation – Decision Procedures with Applications to Verification*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2007.

[BM17] Simone Bova and Fabio Mogavero. Herbrand property, finite quasi-Herbrand models, and a Chandra–Merlin theorem for quantified conjunctive queries. In *Logic in Computer Science (LICS'17)*, pages 1–12, 2017.

[BMR13] Nikolaj Bjørner, Kenneth L. McMillan, and Andrey Rybalchenko. On Solving Universally Quantified Horn Clauses. In *Static Analysis (SAS'13)*, pages 105–125, 2013.

[BMRT11] Jean-François Baget, Marie-Laure Mugnier, Sebastian Rudolph, and Michaël Thomazo. Walking the Complexity Lines for Generalized Guarded Existential Rules. In *Artificial Intelligence (IJCAI'11)*, pages 712–717, 2011.

[BMS06] Aaron R. Bradley, Zohar Manna, and Henny B. Sipma. What's Decidable About Arrays? In *Verification, Model Checking, and Abstract Interpretation (VMCAI'06)*, pages 427–442, 2006.

[BN98] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.

[BPDG98] Béatrice Bérard, Antoine Petit, Volker Diekert, and Paul Gastin. Characterization of the Expressive Power of Silent Transitions in Timed Automata. *Fundamenta Informaticae*, 36(2-3):145–182, 1998.

[Bra07] Aaron R. Bradley. *Safety Analysis of Systems*. PhD thesis, Department of Computer Science of Stanford University, 2007.

[BS28] Paul Bernays and Moses Schönfinkel. Zum Entscheidungsproblem der mathematischen Logik. *Mathematische Annalen*, 99(1):342–372, 1928.

[BT10] Daniel Berend and Tamir Tassa. Improved bounds on Bell numbers and on moments of sums of random variables. *Probability and Mathematical Statistics*, 30(2):185–205, 2010.

[BT18] Clark Barrett and Cesare Tinelli. Satisfiability Modulo Theories. In Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking*, pages 305–343. Springer, 2018.

[BtCS11] Vince Bárány, Balder ten Cate, and Luc Segoufin. Guarded negation. In *Automata, Languages and Programming (ICALP'11), Part II*, pages 356–367, 2011.

[BtCS15] Vince Bárány, Balder ten Cate, and Luc Segoufin. Guarded Negation. *Journal of the ACM*, 62(3):22, 2015.

[BtCV16] Michael Benedikt, Balder ten Cate, and Michael Vanden Boom. Effective Interpolation and Preservation in Guarded Logics. *ACM Transactions on Computational Logic*, 17(2):8:1–8:46, 2016.

[Büc60] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 6:66–92, 1960.

[Büc62] J. Richard Büchi. On a decision method in restricted second order arithmetic. In Ernest Nagel, Patrick Suppes, and Alfred Tarski, editors, *Proceedings of the 1960 International Congress on Logic, Methodology and Philosophy of Science*, pages 1–11. Stanford University Press, 1962.

[BW13a] Peter Baumgartner and Uwe Waldmann. Hierarchic Superposition: Completeness without Compactness. In Marek Košta and Thomas Sturm, editors, *Fifth International Conference on Mathematical Aspects of Computer and Information Sciences (MACIS'13)*, pages 8–12, 2013.

[BW13b] Peter Baumgartner and Uwe Waldmann. Hierarchic superposition with weak abstraction. In *Automated Deduction (CADE-24)*, LNCS 7898, pages 39–57. Springer, 2013.

[Cal02] Cristian Calude. *Information and Randomness : An Algorithmic Perspective*. Springer, second, revised and extended edition, 2002.

[CAMN04] Scott Cotton, Eugene Asarin, Oded Maler, and Peter Niebert. Some Progress in Satisfiability Checking for Difference Logic. In *Formal Modelling and Analysis of Timed Systems and Formal Techniques in Real-Time and Fault-Tolerant Systems (FORMATS/FTRTFT'04)*, pages 263–276, 2004.

[CDG+08] H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree Automata Techniques and Applications. Available on: http://www.grappa.univ-lille3.fr/tata, 2008. Release: November, 18th 2008.

[CFR14] Paula Chocron, Pascal Fontaine, and Christophe Ringeissen. A Gentle Non-Disjoint Combination of Satisfiability Procedures. In *Automated Reasoning (IJCAR'14)*, pages 122–136, 2014.

[CFR15] Paula Chocron, Pascal Fontaine, and Christophe Ringeissen. A Polite Non-Disjoint Combination Method: Theories with Bridging Functions Revisited. In *Automated Deduction (CADE-25)*, pages 419–433, 2015.

[CGP10a] Andrea Calì, Georg Gottlob, and Andreas Pieris. Advanced Processing for Ontological Queries. *Proceedings of VLDB*, 3(1):554–565, 2010.

[CGP10b] Andrea Calì, Georg Gottlob, and Andreas Pieris. Query Answering under Non-guarded Rules in Datalog+/−. In *Web Reasoning and Rule Systems (RR'10)*, pages 1–17, 2010.

[CH90] Kevin J. Compton and C. Ward Henson. A Uniform Method for Proving Lower Bounds on the Computational Complexity of Logical Theories. *Annals of Pure and Applied Logic*, 48(1):1–79, 1990.

[Chu36a] Alonzo Church. Correction to A Note on the Entscheidungsproblem. *Journal of Symbolic Logic*, 1(3):101–102, 1936.

[Chu36b] Alonzo Church. A Note on the Entscheidungsproblem. *Journal of Symbolic Logic*, 1(1):40–41, 1936.

[Chu36c] Alonzo Church. An Unsolvable Problem of Elementary Number Theory. *Journal of Symbolic Logic*, 1(2):73–74, 1936.

[CHVB18] Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem. *Handbook of Model Checking*. Springer, 2018.

[CJ98] Hubert Comon and Yan Jurski. Multiple Counters Automata, Safety Analysis and Presburger Arithmetic. In *Computer Aided Verification (CAV'98)*, pages 268–279, 1998.

[CJ99] Hubert Comon and Yan Jurski. Timed automata and the theory of real numbers. In *Concurrency Theory (CONCUR'99)*, pages 242–257, 1999.

[CK90] Chen Chung Chang and H. Jerome Keisler. *Model Theory*, volume 73 of *Studies in Logic and the Foundations of Mathematics*. Elsevier Science Publishing, third edition, 1990.

[CK06] Sylvain Conchon and Sava Krstic. Strategies for combining decision procedures. *Theoretical Computer Science*, 354(2):187–210, 2006.

[CLM81] Ashok K. Chandra, Harry R. Lewis, and Johann A. Makowsky. Embedded Implicational Dependencies and their Inference Problem. In *Symposium on Theory of Computing (STOC'81)*, pages 342–354, 1981.

[CM93] Jim Cox and Ken McAloon. Decision procedures for constraint-based extensions of Datalog. In Frédéric Benhamou and Alain Colmerauer, editors, *Constraint Logic Programming, Selected Research*, pages 17–32. The MIT Press, 1993.

[CM06] Scott Cotton and Oded Maler. Fast and Flexible Difference Constraint Propagation for DPLL(T). In *Theory and Applications of Satisfiability Testing (SAT'06)*, pages 170–183, 2006.

[CMT92] Jim Cox, Ken McAloon, and Carol Tretkoff. Computational Complexity and Constraint Logic Programming Languages. *Annals of Mathematics and Artificial Intelligence*, 5(2-4):163–189, 1992.

[Con06] Willem Conradie. On the strength and scope of DLS. *Journal of Applied Non-Classical Logics*, 16(3-4):279–296, 2006.

[Coo71] Stephen A. Cook. The Complexity of Theorem-Proving Procedures. In *Theory of Computing (STOC'71)*, pages 151–158, 1971.

[Coo72] David C. Cooper. Theorem Proving in Arithmetic without Multiplication. *Machine Intelligence*, 7:91–99, 1972.

[Coo04] S. Barry Cooper. *Computability Theory*. Chapman & Hall/CRC, 2004.

[Cra57a] William Craig. Linear Reasoning. A New Form of the Herbrand–Gentzen Theorem. *Journal of Symbolic Logic*, 22(3):250–268, 09 1957.

[Cra57b] William Craig. Three Uses of the Herbrand-Gentzen Theorem in Relating Model Theory and Proof Theory. *Journal of Symbolic Logic*, 22(3):269–285, 09 1957.

[CSRL01] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, second edition, 2001.

[CW10] Witold Charatonik and Piotr Witkowski. On the Complexity of the Bernays–Schönfinkel Class with Datalog. In *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-17)*, LNCS 6397, pages 187–201. Springer, 2010.

[Daw99] Anuj Dawar. Finite models and finitely many variables. In D. Niwinski and R. Maron, editors, *Logic, Algebra and Computer Science*, volume 46 of *Banach Center Publications*, pages 93–117. Polish Academy of Sciences, 1999.

[DE73] George B. Dantzig and B. Curtis Eaves. Fourier–Motzkin Elimination and Its Dual. *Journal of Combinatorial Theory, Series A*, 14(3):288–297, 1973.

[DFPP18] Laurent Doyen, Goran Frehse, George J. Pappas, and André Platzer. Verification of Hybrid Systems. In Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking*, pages 1047–1110. Springer, 2018.

[DG79] Burton Dreben and Warren D. Goldfarb. *The Decision Problem: Solvable Classes of Quantificational Formulas*. Addison-Wesley, 1979.

[DGKS07a] Anuj Dawar, Martin Grohe, Stephan Kreutzer, and Nicole Schweikardt. Model Theory Makes Formulas Large. In *Automata, Languages and Programming (ICALP'07)*, pages 913–924, 2007.

[DGKS07b] Anuj Dawar, Martin Grohe, Stephan Kreutzer, and Nicole Schweikardt. Model Theory Makes Formulas Large. Technical Report NI07003-LAA, Isaac Newton Institute of Mathematical Sciences, 2007.

[DH10] Rodney G. Downey and Denis R. Hirschfeldt. *Algorithmic Randomness and Complexity*. Springer, 2010.

[Din19] Lloyd L. Dines. Systems of Linear Inequalities. *Annals of Mathematics, Second Series*, 20(3):191–199, 1919.

[DKW62] Burton Dreben, Andrew Kahr, and Hao Wang. Classification of *AEA* Formulas by Letter Atoms. *Bulletin of the American Mathematical Society*, 68(5):528–532, 1962.

[DL84a] Larry Denenberg and Harry R. Lewis. The Complexity of the Satisfiability Problem for Krom Formulas. *Theoretical Computer Science*, 30:319–341, 1984.

[DL84b] Larry Denenberg and Harry R. Lewis. Logical Syntax and Computational Complexity. In *Computation and Proof Theory. Proceedings of the Logic Colloquium '83, Aachen, Part II*, LNM 1104, pages 101–115. Springer, 1984.

[dMB11] Leonardo Mendonça de Moura and Nikolaj Bjørner. Satisfiability Modulo Theories: Introduction and Applications. *Communications of the ACM*, 54(9):69–77, 2011.

[dN98] Hans de Nivelle. A Resolution Decision Procedure for the Guarded Fragment. In *Automated Deduction (CADE-15)*, pages 191–204, 1998.

[dNdR03] Hans de Nivelle and Maarten de Rijke. Deciding the guarded fragments by resolution. *Journal of Symbolic Computation*, 35(1):21–58, 2003.

[dNP01] Hans de Nivelle and Ian Pratt-Hartmann. A Resolution-Based Decision Procedure for the Two-Variable Fragment with Equality. In *Automated Reasoning (IJCAR'01)*, pages 211–225, 2001.

[DNS05] David Detlefs, Greg Nelson, and James B. Saxe. Simplify: a theorem prover for program checking. *Journal of the ACM*, 52(3):365–473, 2005.

[Dol00] Andreas Dolzmann. *Algorithmic strategies for applicable real quantifier elimination*. PhD thesis, University of Passau, Germany, 2000.

[Dow72] Peter J. Downey. Undecidability of Presburger Arithmetic with a Single Monadic Predicate Letter. Technical report, Center for Research in Computer Technology, Harvard University, 1972.

[Dre62] Burton Dreben. Solvable Surányi subclasses: an introduction to the Herbrand theory. *Annals of the Computation Laboratory of Harvard University*, 31:32–47, 1962.

[EF99] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite model theory. Second Edition*. Perspectives in Mathematical Logic. Springer, 1999.

[EFT94] Heinz-Dieter Ebbinghaus, Jörg Flum, and Wolfgang Thomas. *Mathematical Logic*. Springer, second edition, 1994.

[Egl94] Uwe Egly. On the Value of Antiprenexing. In *Logic Programming and Automated Reasoning (LPAR'94)*, pages 69–83, 1994.

[EKK+11] Andreas Eggers, Evgeny Kruglov, Stefan Kupferschmid, Karsten Scheibler, Tino Teige, and Christoph Weidenbach. Superposition Modulo Non-linear Arithmetic. In *Frontiers of Combining Systems (FroCoS'11)*, pages 119–134, 2011.

[EKK+12] Moshe Emmer, Zurab Khasidashvili, Konstantin Korovin, Christoph Sticksel, and Andrei Voronkov. EPR-Based Bounded Model Checking at Word Level. In *Automated Reasoning (IJCAR'12)*, pages 210–224, 2012.

[EKKV10] Moshe Emmer, Zurab Khasidashvili, Konstantin Korovin, and Andrei Voronkov. Encoding industrial hardware verification problems into effectively propositional logic. In *Formal Methods in Computer-Aided Design (FMCAD'10)*, pages 137–144, 2010.

[End72] Herbert B. Enderton. *A mathematical introduction to logic.* Academic Press, 1972.

[End01] Herbert B. Enderton. *A mathematical introduction to logic.* Harcourt/Academic Press, 2001.

[FA03] Thom Frühwirth and Slim Abdennadher. *Essentials of Constraint Programming.* Springer, 2003.

[FG06] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006.

[Fie13] Arnaud Fietzke. *Labelled Superposition.* PhD thesis, Department of Computer Science, Saarland University, 2013.

[Fit96] Melvin Fitting. *First-Order Logic and Automated Theorem Proving, Second Edition.* Graduate Texts in Computer Science. Springer, 1996.

[FJS04] Cormac Flanagan, Rajeev Joshi, and James B. Saxe. An Explicating Theorem Prover for Quantified Formulas. Technical Report HPL-2004-199, HP Laboratories Palo Alto, 2004.

[FLHT01] Christian G. Fermüller, Alexander Leitsch, Ullrich Hustadt, and Tanel Tammet. Resolution Decision Procedures. In Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, volume II, pages 1791–1849. Elsevier and MIT Press, 2001.

[FLTZ93] Christian G. Fermüller, Alexander Leitsch, Tanel Tammet, and N. K. Zamov. *Resolution Methods for the Decision Problem.* LNCS 679. Springer, 1993.

[FMSZ17] Bernd Finkbeiner, Christian Müller, Helmut Seidl, and Eugen Zalinescu. Verifying Security Policies in Multi-agent Workflows with Loops. In *Computer and Communications Security (CCS'17)*, pages 633–645, 2017.

[Fon07] Pascal Fontaine. Combinations of Theories and the Bernays–Schönfinkel–Ramsey Class. In *Verification Workshop in connection with CADE-21 (VERIFY'07)*, 2007.

[Fon09] Pascal Fontaine. Combinations of Theories for Decidable Fragments of First-Order Logic. In *Frontiers of Combining Systems (FroCoS'09)*, LNCS 5749, pages 263–278. Springer, 2009.

[Fou26] Jean Baptiste Joseph Fourier. Solution d'une Question Particulière du Calcul des Inégalités. *Nouveau Bulletin des Sciences par la Société philomathique de Paris*, pages 99–100, 1826. Reprinted in Jeon Gaston Darboux (editor), *Oeuvres de Fourier*, Tome II, Gauthier-Villars, Paris, 1890, pp. 317–319. Reprinted in 2013 by Cambridge University Press.

[FR74] Michael Jo Fischer and Michael O. Rabin. Super-Exponential Complexity of Presburger Arithmetic. In *SIAM-AMS Symposium in Applied Mathematics*, pages 27–41, 1974.

[FR75] Jeanne Ferrante and Charles Rackoff. A Decision Procedure for the First Order Theory of Real Addition with Order. *SIAM Journal of Computing*, 4(1):69–76, 1975.

[FR79] Jeanne Ferrante and Charles W. Rackoff. *The computational complexity of logical theories*. Springer, 1979.

[FS93] Christian G. Fermüller and Gernot Salzer. Ordered Paramodulation and Resolution as Decision Procedure. In *Logic Programming and Automated Reasoning (LPAR'93)*, pages 122–133, 1993.

[FSVY91] Thom W. Frühwirth, Ehud Y. Shapiro, Moshe Y. Vardi, and Eyal Yardeni. Logic Programs as Types for Logic Programs. In *Logic in Computer Science (LICS'91)*, pages 300–309, 1991.

[Für81] Martin Fürer. Alternation and the Ackermann Case of the Decision Problem. *L'Enseignement Mathématique*, 27(1–2):137–162, 1981.

[Für83] Martin Fürer. The computational complexity of the unconstrained limited domino problem (with implications for logical decision problems). In *Logic and Machines: Decision Problems and Complexity, Proceedings of the Symposium "Rekursive Kombinatorik"*, pages 312–319, 1983.

[FW12] Arnaud Fietzke and Christoph Weidenbach. Superposition as a Decision Procedure for Timed Automata. *Mathematics in Computer Science*, 6(4):409–425, 2012.

[Gai82] Haim Gaifman. On Local and Non-Local Properties. In J. Stern, editor, *Proceedings of the Herbrand Symposium, Logic Colloquium '81*, pages 105–135. North-Holland, 1982.

[Gan02] Harald Ganzinger. Shostak Light. In *Automated Deduction (CADE-18)*, pages 332–346, 2002.

[GBT09] Yeting Ge, Clark W. Barrett, and Cesare Tinelli. Solving quantified verification conditions using satisfiability modulo theories. *Ann. Math. Artif. Intell.*, 55(1-2):101–122, 2009.

[GdM09] Yeting Ge and Leonardo Mendonça de Moura. Complete Instantiation for Quantified Formulas in Satisfiabiliby Modulo Theories. In *Computer Aided Verification (CAV'09)*, LNCS 5643, pages 306–320. Springer, 2009.

[GdN99] Harald Ganzinger and Hans de Nivelle. A Superposition Decision Procedure for the Guarded Fragment with Equality. In *Logic in Computer Science (LICS'99)*, pages 295–303, 1999.

[Gen35a] Gerhard Gentzen. Untersuchungen über das logische Schließen. I. *Mathematische Zeitschrift*, 39(1):176–210, 1935.

[Gen35b] Gerhard Gentzen. Untersuchungen über das logische Schließen. II. *Mathematische Zeitschrift*, 39(1):405–431, 1935.

[GG18] Silvio Ghilardi and Alessandro Gianola. Modularity results for interpolation, amalgamation and superamalgamation. *Annals in Pure and Applied Logic*, 169(8):731–754, 2018.

[GGS84] Warren D. Goldfarb, Yuri Gurevich, and Saharon Shelah. A Decidable Subclass of the Minimal Gödel Class with Identity. *Journal of Symbolic Logic*, 49(4):1253–1261, 1984.

[GHK+13] Bernardo Cuenca Grau, Ian Horrocks, Markus Krötzsch, Clemens Kupke, Despoina Magka, Boris Motik, and Zhe Wang. Acyclicity Notions for Existential Rules and Their Application to Query Answering in Ontologies. *Journal of Artificial Intelligence Research*, 47:741–808, 2013.

[GHN+04] Harald Ganzinger, George Hagen, Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli. DPLL(T): Fast Decision Procedures. In *Computer Aided Verification (CAV'04)*, pages 175–188, 2004.

[GHS00] Lilia Georgieva, Ullrich Hustadt, and Renate A. Schmidt. Hyperresolution for Guarded Formulae. In *Proceedings of the Seventh Workshop on Automated Reasoning, Bridging the Gap between Theory and Practice*, 2000.

[GHS02] Lilia Georgieva, Ullrich Hustadt, and Renate A. Schmidt. A New Clausal Class Decidable by Hyperresolution. In *Automated Deduction (CADE-18)*, pages 260–274, 2002.

[GHS03] Lilia Georgieva, Ullrich Hustadt, and Renate A. Schmidt. Hyperresolution for guarded formulae. *Journal of Symbolic Computation*, 36(1-2):163–192, 2003.

[GHW03] Harald Ganzinger, Thomas Hillenbrand, and Uwe Waldmann. Superposition Modulo a Shostak Theory. In *Automated Deduction (CADE-19)*, pages 182–196, 2003.

[GKL+07] Erich Grädel, Phokion G. Kolaitis, Leonid Libkin, Maarten Marx, Joel Spencer, Moshe Y. Vardi, Yde Venema, and Scott Weinstein. *Finite Model Theory and Its Applications*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2007.

[GKV97] Erich Grädel, Phokion G. Kolaitis, and Moshe Y. Vardi. On the decision problem for two-variable first-order logic. *Bulletin of Symbolic Logic*, 3(1):53–69, 1997.

[GKVV16] Erich Grädel, Juha Kontinen, Jouka Väänänen, and Heribert Vollmer. Logics for Dependence and Independence (Dagstuhl Seminar 15261). *Dagstuhl Reports*, 5(6):70–85, 2016.

[GL81] Péter Gács and László Lovász. Khachiyan's algorithm for linear programming. *Mathematical Programming Study*, (14):61–68, 1981.

[GNZ08] Silvio Ghilardi, Enrica Nicolini, and Daniele Zucchelli. A comprehensive combination framework. *ACM Transactions on Computational Logic*, 9(2):8:1–8:54, 2008.

[GO99] Erich Grädel and Martin Otto. On Logics with Two Variables. *Theoretical Computer Science*, 224(1-2):73–113, 1999.

[Göd32] Kurt Gödel. Ein Spezialfall des Entscheidungsproblems der theoretischen Logik. *Ergebnisse eines mathematischen Kolloquiums*, (2):27–28, 1932. English translation in Solomon Feferman, John W. Dawson, Jr., Stephen C. Kleene, Gregory H. Moore, Robert M. Solovay, and Jean van Heijenoort, *Kurt Gödel: Collected Works: Volume I: Publications 1929-1936*, Oxford University Press, 1986, pp. 230–233.

[Göd33] Kurt Gödel. Zum Entscheidungsproblem des logischen Funktionenkalküls. *Monatshefte für Mathematik und Physik*, 40:433–443, 1933. Reprinted in Solomon Feferman, John W. Dawson, Jr., Stephen C. Kleene, Gregory H. Moore, Robert M. Solovay, and Jean van Heijenoort, *Kurt Gödel: Collected Works: Volume I: Publications 1929-1936*, Oxford University Press, 1986, pp. 306–326.

[Gol63] Richard Goldberg. On the Solvability of a Subclass of the Suranyi Reduction Class. *Journal of Symbolic Logic*, 28(3):237–244, 1963.

[Gol84] Warren D. Goldfarb. The Unsolvability of the Godel Class with Identity. *Journal of Symbolic Logic*, 49(4):1237–1252, 1984.

[Gol08] Oded Goldreich. *Computational Complexity – A Conceptual Perspective*. Cambridge University Press, 2008.

[Gou95] Jean Goubault. A BDD-Based Simplification and Skolemization Procedure. *Logic Journal of the IGPL*, 3(6):827–855, 1995.

[Gou05] Jean Goubault-Larrecq. Deciding $\mathcal{H}_1$ by resolution. *Information Processing Letters*, 95(3):401–408, 2005.

[Grä90a] Erich Grädel. On solvable cases of Hilbert's 'Entscheidungsproblem'. Habilitationsschrift, Universität Basel, 1990.

[Grä90b] Erich Grädel. Satisfiability of Formulae with One $\forall$ is Decidable in Exponential Time. *Archive for Mathematical Logic*, 29:265–276, 1990.

[Grä99a] Erich Grädel. Invited Talk: Decision procedures for guarded logics. In *Automated Deduction (CADE-16)*, pages 31–51, 1999.

[Grä99b] Erich Grädel. On the Restraining Power of Guards. *Journal of Symbolic Logic*, 64:1719–1742, 12 1999.

[Gro98] Martin Grohe. Finite variable logics in descriptive complexity theory. *Bulletin of Symbolic Logic*, 4(4):345–398, 1998.

[GRS90] R.L. Graham, B.L. Rothschild, and J.H. Spencer. *Ramsey Theory*. A Wiley-Interscience publication. Wiley, second edition, 1990.

[GS74] Solomon Garfunkel and James H Schmerl. The undecidability of theories of groupoids with an extra predicate. *Proceedings of the American Mathematical Society*, 42(1):286–289, 1974.

[GS83] Yuri Gurevich and Saharon Shelah. Random Models and the Gödel Case of the Decision Problem. *Journal of Symbolic Logic*, 48(4):1120–1124, 1983.

[GSS08] Dov Gabbay, Renate Schmidt, and Andrzej Szałas. *Second-Order Quantifier Elimination: Foundations, Computational Aspects and Applications*. College Publications, 2008.

[Gur69] Yuri Gurevich. The Decision Problem for the Logic of Predicates and Operations. *Algebra i Logika*, 8:284–308, 1969.

[Gur73] Y. Gurevich. Formulas with one $\forall$. In *Selected Questions in Algebra and Logic; in memory of A. Malcev*, pages 97–110. Nauka, Moscow, 1973. In Russian. A German translation is available at TIB Universität Hannover, Germany.

[Gur76] Yuri Gurevich. The Decision Problem for Standard Classes. *Journal of Symbolic Logic*, 41(2):460–464, 1976.

[HA28] David Hilbert and Wilhelm Ackermann. *Grundzüge der theoretischen Logik*. Springer, 1928.

[Haa18] Christoph Haase. A Survival Guide to Presburger Arithmetic. *SIGLOG News*, 5(3):67–82, 2018.

[Hal91] Joseph Y. Halpern. Presburger Arithmetic with Unary Predicates is Π^1_1 Complete. *Journal of Symbolic Logic*, 56(2):637–642, 1991.

[Har87] Juris Hartmanis. The Structural Complexity Column: The Collapsing Hierarchies. *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, (33):26–39, 1987.

[Hen61] Leon Henkin. Some remarks on infinitely long formulas. In *Infinistic Methods*, pages 167–183. Pergamon Press, 1961.

[Her30] Jacques Herbrand. *Recherches sur la théorie de la démonstration*. PhD thesis, L'Université de Paris, 1930. English translation in Jacques Herbrand, *Logical Writings* (edited by Warren D. Goldfarb), D. Reidel Publishing Company, 1971.

[Her90] Andreas Herzig. A new decidable fragment of first order logic. In *Abstracts of the Third Logical Biennial, Summer School & Conference in Honour of S. C. Kleene, Varna, Bulgaria*, 1990.

[Hil08] Thomas Hillenbrand. *Superposition and Decision Procedures Back and Forth*. PhD thesis, Department of Computer Science, Saarland University, 2008.

[Hin65] Jaakko Hintikka. Distributive Normal Forms in First-Order Logic. In J.N. Crossley and M.A.E. Dummett, editors, *Formal Systems and Recursive Functions*, volume 40 of *Studies in Logic and the Foundations of Mathematics*, pages 48–91. Elsevier, 1965.

[Hin73] Jaakko Hintikka. *Logic, Language-Games and Information: Kantian Themes in the Philosophy of Logic*. Clarendon Press, 1973.

[HIV08] Peter Habermehl, Radu Iosif, and Tomás Vojnar. What Else Is Decidable about Integer Arrays? In *Foundations of Software Science and Computational Structures (FOSSACS'08)*, pages 474–489, 2008.

[HM99] Eva Hoogland and Maarten Marx. Interpolation in Guarded Fragments. Technical report, Institute for Logic, Language and Computation (ILLC), University of Amsterdam, 1999.

[HM02] Eva Hoogland and Maarten Marx. Interpolation and Definability in Guarded Fragments. *Studia Logica*, 70(3):373–409, 2002.

[HMO99] Eva Hoogland, Maarten Marx, and Martin Otto. Beth Definability for the Guarded Fragment. In *Logic Programming and Automated Reasoning (LPAR'99)*, pages 273–285, 1999.

[HMU01] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, second edition, 2001.

[HNSY94] Thomas A. Henzinger, Xavier Nicollin, Joseph Sifakis, and Sergio Yovine. Symbolic Model Checking for Real-Time Systems. *Information and Computation*, 111(2):193–244, 1994.

[Hod93] Wilfrid Hodges. *Model theory*. Cambridge University Press, 1993.

[Hod02] Ian M. Hodkinson. Loosely Guarded Fragment of First-Order Logic has the Finite Model Property. *Studia Logica*, 70(2):205–240, 2002.

[HOD17] Ullrich Hustadt, Ana Ozaki, and Clare Dixon. Theorem Proving for Metric Temporal Logic over the Naturals. In *Automated Deduction (CADE-26)*, pages 326–343, 2017.

[Hoo01] Eva Hoogland. *Definability and Interpolation: Model-theoretic investigations*. PhD thesis, Institute for Logic, Language and Computation, University of Amsterdam, 2001.

[HPS83] David Harel, Amir Pnueli, and Jonathan Stavi. Propositional dynamic logic of nonregular programs. *Journal of Computer and System Sciences*, 26(2):222–243, 1983.

[HS99] Ullrich Hustadt and Renate A. Schmidt. Maslov's Class K Revisited. In *Automated Deduction (CADE-16)*, pages 172–186, 1999.

[HS18] Youssef Hamadi and Lakhdar Sais. *Handbook of Parallel Constraint Reasoning*. Springer, 2018.

[HSBW13] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

[HSG04] Ullrich Hustadt, Renate A. Schmidt, and Lilia Georgieva. A survey of decidable first-order fragments and description logics. *Journal on Relational Methods in Computer Science*, 1:251–276, 2004.

[HU79] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.

[Hua95] Guoxiang Huang. Constructing Craig Interpolation Formulas. In *Computing and Combinatorics (COCOON'95)*, pages 181–190, 1995.

[Hus99] Ullrich Hustadt. *Resolution-Based Decision Procedures for Subclasses of First-Order Logic*. PhD thesis, Saarland University, Saarbrücken, Germany, 1999.

[HVW17a] Matthias Horbach, Marco Voigt, and Christoph Weidenbach. On the Combination of the Bernays–Schönfinkel–Ramsey Fragment with Simple Linear Integer Arithmetic. In *Automated Deduction (CADE'17)*, pages 77–94, 2017. An extended version is available at the arXiv preprint server (`arXiv.org`) under the signature arXiv:1705.08792 [cs.LO].

[HVW17b] Matthias Horbach, Marco Voigt, and Christoph Weidenbach. The Universal Fragment of Presburger Arithmetic with Unary Uninterpreted Predicates is Undecidable. *ArXiv preprint*, (arXiv:1703.01212 [cs.LO]), 2017.

[HW13] Thomas Hillenbrand and Christoph Weidenbach. Superposition for Bounded Domains. In *Automated Reasoning and Mathematics – Essays in Memory of William W. McCune*, LNCS 7788, pages 68–100. Springer, 2013.

[IBI+13] Shachar Itzhaky, Anindya Banerjee, Neil Immerman, Aleksandar Nanevski, and Mooly Sagiv. Effectively-Propositional Reasoning about Reachability in Linked Data Structures. In *Computer Aided Verification (CAV'13)*, pages 756–772, 2013.

[IBI+14] Shachar Itzhaky, Anindya Banerjee, Neil Immerman, Ori Lahav, Aleksandar Nanevski, and Mooly Sagiv. Modular reasoning about heap paths via effectively propositional formulas. In *Principles of Programming Languages (POPL'14)*, pages 385–396, 2014.

[IBR+14] Shachar Itzhaky, Nikolaj Bjørner, Thomas W. Reps, Mooly Sagiv, and Aditya V. Thakur. Property-Directed Shape Analysis. In *Computer Aided Verification (CAV'14)*, pages 35–51, 2014.

[Imm88] Neil Immerman. Nondeterministic Space is Closed Under Complementation. *SIAM Journal on Computing*, 17(5):935–938, 1988.

[JL77] Neil D. Jones and William T. Laaser. Complete Problems for Deterministic Polynomial Time. *Theoretical Computer Science*, 3:105–117, 1977.

[JLL76] Neil D. Jones, Y. Edmund Lien, and William T. Laaser. New problems complete for nondeterministic log space. *Mathematical systems theory*, 10(1):1–17, 1976.

[JMW98] Florent Jacquemard, Christoph Meyer, and Christoph Weidenbach. Unification in Extensions of Shallow Equational Theories. In *Rewriting Techniques and Applications (RTA'98)*, pages 76–90, 1998.

[Joy76] William H. Joyner Jr. Resolution Strategies as Decision Procedures. *Journal of the ACM*, 23(3):398–417, 1976.

[JRV06] Florent Jacquemard, Michaël Rusinowitch, and Laurent Vigneron. Tree Automata with Equality Constraints Modulo Equational Theories. In *Automated Reasoning (IJCAR'06)*, pages 557–571, 2006.

[Kal33] László Kalmár. Über die Erfüllbarkeit derjenigen Zählausdrücke, welche in der Normalform zwei benachbarte Allzeichen enthalten. *Mathematische Annalen*, 108:466–484, 1933.

[Kar84] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

[Kas86] Simon Kasif. On the Parallel Complexity of Some Constraint Satisfaction Problems. In *Artificial Intelligence (AAAI'86)*, pages 349–353, 1986.

[KBI+15] Aleksandr Karbyshev, Nikolaj Bjørner, Shachar Itzhaky, Noam Rinetzky, and Sharon Shoham. Property-Directed Inference of Universal Invariants or Proving Their Absence. In *Computer Aided Verification (CAV'15)*, pages 583–602, 2015.

[Kha80] Leonid Gendrichowitsch Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53–72, 1980.

[KK14] Emanuel Kieronski and Antti Kuusisto. Complexity and Expressivity of Uniform One-Dimensional Fragment with Equality. In *Mathematical Foundations of Computer Science (MFCS'14)*, pages 365–376, 2014.

[KM95] Michał Krynicki and Marcin Mostowski. Henkin Quantifiers. In Michał Krynicki, Marcin Mostowski, and Lesław W. Szczerba, editors, *Quantifiers: Logics, Models and Computation*, pages 193–263. Kluwer Academic Publishers, 1995.

[Kor13a] Konstantin Korovin. Inst-Gen – A Modular Approach to Instantiation-Based Automated Reasoning. In *Programming Logics – Essays in Memory of Harald Ganzinger*, pages 239–270, 2013.

[Kor13b] Konstantin Korovin. Non-cyclic Sorts for First-Order Satisfiability. In *Frontiers of Combining Systems (FroCoS'13)*, LNCS 8152, pages 214–228. Springer, 2013.

[Koš16] Marek Košta. *New Concepts for Real Quantifier Elimination by Virtual Substitution*. PhD thesis, Department of Computer Science, Saarland University, 2016.

[KPHT18] Emanuel Kieroński, Ian Pratt-Hartmann, and Lidia Tendera. Two-variable logics with counting and semantic constraints. *SIGLOG News*, 5(3):22–43, 2018.

[KPSW10] Viktor Kuncak, Ruzica Piskac, Philippe Suter, and Thomas Wies. Building a Calculus of Data Structures. In *Verification, Model Checking, and Abstract Interpretation (VMCAI'10)*, pages 26–44, 2010.

[KR11] Markus Krötzsch and Sebastian Rudolph. Extending Decidable Existential Rules by Joining Acyclicity and Guardedness. In *Artificial Intelligence (IJCAI'11)*, pages 963–968, 2011.

[Kro67] Melven R. Krom. The Decision Problem for Segregated Formulas in First-Order Logic. *Mathematica Scandinavica*, 21:233–240, 1967.

[KRSW17] Patrick Koopmann, Sebastian Rudolph, Renate A. Schmidt, and Christoph Wernhard, editors. *Proceedings of the Workshop on Second-Order Quantifier Elimination and Related Topics (SOQE 2017)*, volume 2013 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017.

[Kru13] Evgeny Kruglov. *Superposition Modulo Theory*. PhD thesis, Department of Computer Science, Saarland University, 2013.

[KS10] Konstantin Korovin and Christoph Sticksel. iProver-Eq: An Instantiation-Based Theorem Prover with Equality. In *Automated Reasoning (IJCAR'10)*, pages 196–202, 2010.

[KS16] Daniel Kroening and Ofer Strichman. *Decision Procedures*. Texts in Theoretical Computer Science. An EATCS Series. Springer, second edition, 2016.

[Kue71] D. G. Kuehner. A note on the relation between resolution and Maslov's inverse method. In *Machine Intelligence 6*, chapter 5, pages 73–76. Edinburgh University Press, 1971.

[KV90] Phokion G. Kolaitis and Moshe Y. Vardi. 0-1 Laws and Decision Problems for Fragments of Second-Order Logic. *Information and Computation*, 87(1/2):301–337, 1990.

[KV17] Laura Kovács and Andrei Voronkov. First-Order Interpolation and Interpolating Proof Systems. In *Logic for Programming, Artificial Intelligence and Reasoning (LPAR'17)*, pages 49–64, 2017.

[KW12] Evgeny Kruglov and Christoph Weidenbach. Superposition Decides the First-Order Logic Fragment Over Ground Theories. *Mathematics in Computer Science*, 6(4):427–456, 2012.

[Lei93] Alexander Leitsch. Deciding Clause Classes by Semantic Clash Resolution. *Fundamenta Informaticae*, 18:163–182, 1993.

[Lei97] Alexander Leitsch. *The Resolution Calculus*. Texts in theoretical computer science. An EATCS Series. Springer, 1997.

[Lei99] Alexander Leitsch. Resolution and the Decision Problem. In Andrea Cantini, Ettore Casari, and Pierluigi Minari, editors, *Logic and Foundations of Mathematics: Selected Contributed Papers of the Tenth International Congress of Logic, Methodology and Philosophy of Science, Florence, August 1995*, pages 249–269. Springer Netherlands, 1999.

[Lev73] Leonid A. Levin. Universal Sequential Search Problems. *Problemy peredachi informatsii*, 9(3):115–116, 1973. An English translation can be found in the appendix in [Tra84].

[Lew78] Harry R. Lewis. Complexity of Solvable Cases of the Decision Problem for the Predicate Calculus. In *Foundations of Computer Science (FOCS'78)*, pages 35–47, 1978.

[Lew79] Harry R. Lewis. *Unsolvable Classes of Quantificational Formulas*. Addison-Wesley, 1979.

[Lew80] Harry R. Lewis. Complexity Results for Classes of Quantificational Formulas. *Journal of Computer and System Sciences*, 21(3):317–353, 1980.

[Lew90] Harry R. Lewis. A Logic of Concrete Time Intervals (Extended Abstract). In *Logic in Computer Science (LICS'90)*, pages 380–389, 1990.

[Lib04] Leonid Libkin. *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.

[Lif69] V. A. Lifshits. Some Reduction Classes and Undecidable Theories. In *Studies in Constructive Mathematics and Mathematical Logic*, volume 4 of *Seminars in Mathematics*, pages 24–25. Steklov Mathematical Institute, 1969.

[Lif89] Vladimir Lifschitz. What Is the Inverse Method? *Journal of Automated Reasoning*, 5(1):1–23, 1989.

[LMTV12] Nicola Leone, Marco Manna, Giorgio Terracina, and Pierfrancesco Veltri. Efficiently Computable Datalog∃ Programs. In *Knowledge Representation and Reasoning (KR'12)*, 2012.

[Löb67] Martin H. Löb. Decidability of the monadic predicate calculus with unary function symbols. *Journal of Symbolic Logic*, 32:563, 1967.

[Lon12] Florian Lonsing. *Dependency Schemes and Search-Based QBF Solving: Theory and Practice*. PhD thesis, Johannes–Keppler–Universität Linz, 2012.

[Łoś55] Jerzy Łoś. On the extending of models (I). *Fundamenta mathematicae*, 42:38–54, 1955.

[Löw15] Leopold Löwenheim. Über Möglichkeiten im Relativkalkül. *Mathematische Annalen*, 76:447–470, 1915. English translation in [vH02].

[LW93] Rüdiger Loos and Volker Weispfenning. Applying Linear Quantifier Elimination. *The Computer Journal*, 36(5):450–462, 1993.

[LW13] Manuel Lamotte-Schubert and Christoph Weidenbach. BDI: A New Decidable First-order Clause Class. In *Logic for Programming, Artificial Intelligence and Reasoning (LPAR-19)*, EPiC 26, pages 62–74. EasyChair, 2013.

[LW17] Manuel Lamotte-Schubert and Christoph Weidenbach. BDI: a new decidable clause class. *Journal of Logic and Computation*, 27(2):441–468, 2017.

[Lyn59] Roger C. Lyndon. An interpolation theorem in the predicate calculus. *Pacific Journal of Mathematics*, 9(1):129–142, 1959.

[Mah03] Moez Mahfoudh. *Sur la Vérification de la Satisfaction pour la Logique des Différences*. PhD thesis, Université Joseph Fourier – Grenoble 1, 2003.

[Mar01] Maarten Marx. Tolerance Logic. *Journal of Logic, Language and Information*, 10(3):353–374, 2001.

[Mas64] Sergei Yu. Maslov. An Inverse Method of Establishing Deducibilities in the Classical Predicate Calculus. *Doklady Akademii Nauk SSSR*, 159:1420–1424, 1964.

[Mas68] Sergei Yu. Maslov. The Inverse Method for Establishing Deducibility for Logical Calculi (in Russian). *Trudy Matem. Inst. AN SSSR*, 98:26–87, 1968.
English translation in *Proceedings of the Steklov Institute of Mathematics* 98:25–95, 1968. American Mathematical Society, 1971.

[Mey74] Albert R. Meyer. The Inherent Computational Complexity of Theories of Ordered Sets. In *Proceedings of the International Congress of Mathematicians*, pages 477–482, 1974.

[Min67] Marvin Lee Minsky. *Computation: finite and infinite machines*. Prentice-Hall, 1967.

[MNAM02] Moez Mahfoudh, Peter Niebert, Eugene Asarin, and Oded Maler. A Satisfiability Checker for Difference Logic. In *Theory and Applications of Satisfiability Testing (SAT'02)*, pages 222–230, 2002.

[MO72] Sergei Yu. Maslov and V. P. Orevkov. Decidable classes reducing to a one-quantifier class. *Trudy Matem. Inst. AN SSSR*, 121, 1972. In Russian. English translation in *Proceedings of the Steklov Institute of Mathematics* 121:61–72, 1972. American Mathematical Society, 1974.

[Mor75] Michael Mortimer. On Languages with Two Variables. *Mathematical Logic Quarterly*, 21(1):135–140, 1975.

[Mot36] T. S. Motzkin. *Beiträge zur Theorie der Linearen Ungleichungen*. PhD thesis, University of Basel, 1936.

[MP15] Fabio Mogavero and Giuseppe Perelli. Binding Forms in First-Order Logic. In *Computer Science Logic (CSL'15)*, pages 648–665, 2015.

[Mug11] Marie-Laure Mugnier. Ontological Query Answering with Existential Rules. In *Web Reasoning and Rule Systems (RR'11)*, pages 2–23, 2011.

[MZ02] Zohar Manna and Calogero G. Zarba. Combining Decision Procedures. In *Formal Methods at the Crossroads. From Panacea to Foundational Support, 10th Anniversary Colloquium of UNU/IIST, the International Institute for Software Technology of The United Nations University, Lisbon, Portugal, Revised Papers*, pages 381–422, 2002.

[NDFK12] Timothy Nelson, Daniel J. Dougherty, Kathi Fisler, and Shriram Krishnamurthi. Toward a More Complete Alloy. In *Abstract State Machines, Alloy, B, VDM, and Z (ABZ'12)*, pages 136–149, 2012.

[Nel84] Greg Nelson. Combining satisfiability procedures by equality-sharing. *Contemporary Mathematics*, 29:201–211, 1984.

[Nie96] Robert Nieuwenhuis. Basic Paramodulation and Decidable Theories (Extended Abstract). In *Logic in Computer Science (LICS'96)*, pages 473–482, 1996.

[NMA+02] Peter Niebert, Moez Mahfoudh, Eugene Asarin, Marius Bozga, Oded Maler, and Navendu Jain. Verification of Timed Automata via Satisfiability Checking. In *Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT'02)*, pages 225–244, 2002.

[NNS02] Flemming Nielson, Hanne Riis Nielson, and Helmut Seidl. Normalizable Horn Clauses, Strongly Recognizable Relations, and Spi. In *Static Analysis (SAS'02)*, pages 20–35, 2002.

[NO79] Greg Nelson and Derek C. Oppen. Simplification by Cooperating Decision Procedures. *ACM Transactions on Programming Languages and Systems*, 1(2):245–257, 1979.

[NO05] Robert Nieuwenhuis and Albert Oliveras. DPLL(T) with Exhaustive Theory Propagation and Its Application to Difference Logic. In *Computer Aided Verification (CAV'05)*, pages 321–334, 2005.

[Noa80] Aris Noah. Predicate-functors and the limits of decidability in logic. *Notre Dame Journal of Formal Logic*, 21(4):701–707, 1980.

[NW01] Andreas Nonnengart and Christoph Weidenbach. Computing Small Clause Normal Forms. In Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, volume I, pages 335–367. Elsevier and MIT Press, 2001.

[Odi92] Piergiorgio Odifreddi. *Classical Recursion Theory*, volume 125 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 1992.

[Opp80] Derek C. Oppen. Complexity, Convexity and Combinations of Theories. *Theoretical Computer Science*, 12:291–302, 1980.

[Ott97] Martin Otto. *Bounded Variable Logics and Counting: A Study in Finite Models*, volume 9 of *Lecture Notes in Logic*. Springer, 1997. Reprinted by Cambridge University Press in 2017.

[Ott00] Martin Otto. An interpolation theorem. *Bulletin of Symbolic Logic*, 6(4):447–462, 2000.

[OW08] Joël Ouaknine and James Worrell. Some Recent Results in Metric Temporal Logic. In *Formal Modeling and Analysis of Timed Systems (FORMATS'08)*, pages 1–13, 2008.

[Pap94] Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.

[PdMB10] Ruzica Piskac, Leonardo Mendonça de Moura, and Nikolaj Bjørner. Deciding Effectively Propositional Logic Using DPLL and Substitution Sets. *Journal of Automated Reasoning*, 44(4):401–424, 2010.

[Pla84] David A. Plaisted. Complete Problems in the First-Order Predicate Calculus. *Journal of Computer and System Sciences*, 29(1):8–35, 1984.

[PMP+16] Oded Padon, Kenneth L. McMillan, Aurojit Panda, Mooly Sagiv, and Sharon Shoham. Ivy: safety verification by interactive generalization. In *Programming Language Design and Implementation (PLDI'16)*, pages 614–630, 2016.

[PO12] Alberto Policriti and Eugenio Omodeo. The Bernays–Schönfinkel–Ramsey class for set theory: decidability. *Journal of Symbolic Logic*, 77:896–918, 2012.

[Pra77] Vaughan R. Pratt. Two Easy Theories Whose Combination is Hard. Technical report, Massachusetts Institute of Technology, 1977.

[Pre29] Mojżesz Presburger. Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In *Sprawozdanie z I Kongresu matematyków krajów słowiańskich, Warszawa*, pages 92–101, 1929. See [Sta84] for an English translation.

[PST16] Ian Pratt-Hartmann, Wieslaw Szwast, and Lidia Tendera. Quine's Fluted Fragment is Non-Elementary. In *Computer Science Logic (CSL'16)*, pages 39:1–39:21, 2016.

[Pur96a] William C. Purdy. Decidability of Fluted Logic with Identity. *Notre Dame Journal of Formal Logic*, 37(1):84–104, 1996.

[Pur96b] William C. Purdy. Fluted Formulas and the Limits of Decidability. *Journal of Symbolic Logic*, 61(2):608–620, 1996.

[Pur99] William C. Purdy. Quine's 'Limits of Decision'. *Journal of Symbolic Logic*, 64(4):1439–1466, 1999.

[Pur02] William C. Purdy. Complexity and Nicety of Fluted Logic. *Studia Logica*, 71(2):177–198, 2002.

[Put57] Hilary Putnam. Decidability and Essential Undecidability. *Journal of Symbolic Logic*, 22(1):39–54, 1957.

[PV07a] Juan Antonio Navarro Pérez and Andrei Voronkov. Encodings of Bounded LTL Model Checking in Effectively Propositional Logic. In *Automated Deduction (CADE-21)*, pages 346–361, 2007.

[PV07b] Juan Antonio Navarro Pérez and Andrei Voronkov. Encodings of Problems in Effectively Propositional Logic. In *Theory and Applications of Satisfiability Testing (SAT'07)*, page 3, 2007.

[PV08] Juan Antonio Navarro Pérez and Andrei Voronkov. Proof Systems for Effectively Propositional Logic. In *Automated Reasoning (IJCAR'08)*, pages 426–440, 2008.

[PV13] Juan Antonio Navarro Pérez and Andrei Voronkov. Planning with Effectively Propositional Logic. In *Programming Logics – Essays in Memory of Harald Ganzinger*, pages 302–316, 2013.

[QSW17] Karin Quaas, Mahsa Shirmohammadi, and James Worrell. Revisiting reachability in timed automata. In *Logic in Computer Science (LICS'17)*, pages 1–12, 2017.

[Qui69] Willard Van Orman Quine. On the Limits of Decision. In *14th International Congress of Philosophy*, volume III, pages 57–62, 1969. An extended version appeared in W.V.Quine *Theories and Things*, Harvard University Press, 1981.

[Qui76] Willard Van Orman Quine. The Variable. In Willard Van Orman Quine, editor, *The Ways of Paradox and other essays. Revised and enlarged edition.* Harvard University Press, 1976.

[Rab69] Michael O. Rabin. Decidability of Second-Order Theories and Automata on Infinite Trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.

[Rac75] Charles W. Rackoff. The complexity of theories of the monadic predicate calculus. Technical Report , IRIA Report 136, 1975.

[Ram30] Frank Plumpton Ramsey. On a Problem of Formal Logic. *Proceedings of The London Mathematical Society*, s2-30:264–286, 1930.

[Ran87] Veikko Rantala. Constituents. In Radu J. Bogdan, editor, *Jaakko Hintikka*, pages 43–76. Springer Netherlands, 1987.

[RBF18] Andrew Reynolds, Haniel Barbosa, and Pascal Fontaine. Revisiting Enumerative Instantiation. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS'18)*, pages 112–131, 2018.

[RESW14] Salvatore Ruggieri, Pavlos Eirinakis, K. Subramani, and Piotr J. Wojciechowski. On the complexity of quantified linear systems. *Theoretical Computer Science*, 518:128–134, 2014.

[RIS17] Andrew Reynolds, Radu Iosif, and Cristina Serban. Reasoning in the Bernays–Schönfinkel–Ramsey Fragment of Separation Logic. In *Verification, Model Checking, and Abstract Interpretation (VMCAI'17)*, pages 462–482, 2017.

[RK15] Andrew Reynolds and Viktor Kuncak. Induction for SMT Solvers. In *Verification, Model Checking, and Abstract Interpretation (VMCAI'15)*, pages 80–98, 2015.

[RKK17] Andrew Reynolds, Tim King, and Viktor Kuncak. Solving quantified linear arithmetic by counterexample-guided instantiation. *Formal Methods in System Design*, 51(3):500–532, 2017.

[Rob49] Julia Robinson. Definability and Decision Problems in Arithmetic. *Journal of Symbolic Logic*, 14(2):98–114, 1949.

[Rog87] Hartley Rogers. *Theory of recursive functions and effective computability*. MIT Press, 1987. This is a paperback reprint of the 1967 original with the same author and title published by McGraw-Hill.

[RRT04] Silvio Ranise, Christophe Ringeissen, and Duc-Khanh Tran. Nelson–Oppen, Shostak and the Extended Canonizer: A Family Picture with a Newborn. In *Theoretical Aspects of Computing (ICTAC'04), Revised Selected Papers*, pages 372–386, 2004.

[RRZ05] Silvio Ranise, Christophe Ringeissen, and Calogero G. Zarba. Combining Data Structures with Nonstably Infinite Theories Using Many-Sorted Logic. In *Frontiers of Combining Systems (FroCoS'05)*, pages 48–64, 2005.

[RS01] Harald Rueß and Natarajan Shankar. Deconstructing Shostak. In *Logic in Computer Science (LICS'01)*, pages 19–28, 2001.

[RSH+16] Thomas Rebele, Fabian M. Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. YAGO: A Multilingual Knowledge Base from Wikipedia, Wordnet, and Geonames. In *The Semantic Web (ISWC'16)*, pages 177–185, 2016.

[RTdM14] Andrew Reynolds, Cesare Tinelli, and Leonardo Mendonça de Moura. Finding conflicting instances of quantified formulas in SMT. In *Formal Methods in Computer-Aided Design (FMCAD'14)*, pages 195–202, 2014.

[RV01] John Alan Robinson and Andrei Voronkov. *Handbook of Automated Reasoning. I & II*. North Holland, 2001.

[Sam08] Marko Samer. Variable Dependencies of Quantified CSPs. In *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'08)*, pages 512–527, 2008.

[Sav70] Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970.

[SBTW18] Anders Schlichtkrull, Jasmin Christian Blanchette, Dmitriy Traytel, and Uwe Waldmann. Formalizing Bachmair and Ganzinger's Ordered Resolution Prover. In *Automated Reasoning (IJCAR'18)*, pages 89–107, 2018.

[SC10] Abhisekh Sankaran and Supratik Chakraborty. On Semantic Generalizations of the Bernays-Schönfinkel-Ramsey Class with Finite or Co-finite Spectra. *ArXiv preprint*, arXiv:1002.4334 [cs.LO], 2010.

[Sch34a] Kurt Schütte. Über die Erfüllbarkeit einer Klasse von logischen Formeln. *Mathematische Annalen*, 110(2):161–194, 1934.

[Sch34b] Kurt Schütte. Untersuchungen zum Entscheidungsproblem der mathematischen Logik. *Mathematische Annalen*, 109(4):572–603, 1934.

[Sch99] Alexander Schrijver. *Theory of linear and integer programming*. Wiley-Interscience series in discrete mathematics and optimization. Wiley, 1999.

[Sch08] Uwe Schöning. *Logic for Computer Scientists*. Birkhäuser, 2008.

[Sch16] Sylvain Schmitz. Complexity Hierarchies beyond Elementary. *ACM Transactions on Computation Theory*, 8(1):3:1–3:36, 2016.

[Sco62] Dana Scott. A decision method for validity of sentences in two variables. *Journal of Symbolic Logic*, 27:477, 1962.

[Seg17] Luc Segoufin. A survey on guarded negation. *SIGLOG News*, 4(3):12–26, 2017.

[SH00] Renate A. Schmidt and Ullrich Hustadt. A Resolution Decision Procedure for Fluted Logic. In *Automated Deduction (CADE-17)*, pages 433–448, 2000.

[She75] Saharon Shelah. The Monadic Theory of Order. *Annals of Mathematics*, 102(3):379–419, 1975.

[She77] Saharon Shelah. Decidability of a portion of the predicate calculus. *Israel Journal on Mathematics*, 28:32–44, 1977.

[Sho84] Robert E. Shostak. Deciding Combinations of Theories. *Journal of the ACM*, 31(1):1–12, 1984.

[Sko19] Thoralf Skolem. Untersuchungen über die Axiome des Klassenkalküls und über Produktations- und Summationsprobleme welche gewisse Klassen von Aussagen betreffen. *Videnskapsselskapets Skrifter I. Mat.-Nat. Klasse (3)*, 1919.

[Sko35] Thoralf Skolem. Ein Satz über Zählausdrücke. *ACTA Scientarium Mathematicarum*, 7:193–199, 1935.

[SKW08] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. YAGO: A Large Ontology from Wikipedia and WordNet. *Journal of Web Semantics*, 6(3):203–217, 2008.

[SM75] Jürgen Schulte-Mönting. Interpolation formulae for predicates and terms which carry their own history. *Archive for Mathematical Logic*, 17(3-4):159–169, 1975.

[Smu95] Raymond M. Smullyan. *First-Order Logic*. Dover Publications, 1995.

[Soa87] Robert Irving Soare. *Recursively Enumerable Sets and Degrees*. Springer, 1987.

[Soa16] Robert Irving Soare. *Turing Computability Theory and Applications*. Theory and Applications of Computability, In cooperation with the association Computability in Europe. Springer, 2016.

[Sof13] Viorica Sofronie-Stokkermans. On Combinations of Local Theory Extensions. In *Programming Logics – Essays in Memory of Harald Ganzinger*, pages 392–413, 2013.

[Sof14] Viorica Sofronie-Stokkermans. Hierarchical Reasoning in Local Theory Extensions and Applications. In *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'14)*, pages 34–41, 2014.

[Spe13a] Stanislav O. Speranski. Collapsing probabilistic hierarchies. I. *Algebra and Logic*, 52(2):159–171, 2013.

[Spe13b] Stanislav O. Speranski. A note on definability in fragments of arithmetic with free unary predicates. *Archive for Mathematical Logic*, 52(5-6):507–516, 2013.

[SR02] Natarajan Shankar and Harald Rueß. Combining Shostak Theories. In *Rewriting Techniques and Applications (RTA'02)*, pages 1–18, 2002.

[SR11] Helmut Seidl and Andreas Reuß. Extending H_1-clauses with disequalities. *Information Processing Letters*, 111(20):1007–1013, 2011.

[SR12] Helmut Seidl and Andreas Reuß. Extending $\mathcal{H}_1$-Clauses with Path Disequalities. In *Foundations of Software Science and Computational Structures (FOSSACS'12)*, pages 165–179, 2012.

[SS06] Geoff Sutcliffe and Christian B. Suttner. The state of CASC. *AI Communications*, 19(1):35–48, 2006.

[SS09] Marko Samer and Stefan Szeider. Backdoor Sets of Quantified Boolean Formulas. *Journal of Automated Reasoning*, 42(1):77–97, 2009.

[SSB02] Ofer Strichman, Sanjit A. Seshia, and Randal E. Bryant. Deciding Separation Formulas with SAT. In *Computer Aided Verification (CAV'02)*, pages 209–222, 2002.

[ST08] Renate A. Schmidt and Dmitry Tishkovsky. A General Tableau Method for Deciding Description Logics, Modal Logics and Related First-Order Fragments. In *Automated Reasoning (IJCAR'08)*, pages 194–209, 2008.

[Sta84] Ryan Stansifer. Presburger's Article on Integer Arithmetic: Remarks and Translation. Technical Report TR84-639, Cornell University, Computer Science Department, 1984.

[StC13] Luc Segoufin and Balder ten Cate. Unary negation. *Logical Methods in Computer Science*, 9(3), 2013.

[Stu17] Thomas Sturm. A Survey of Some Methods for Real Quantifier Elimination, Decision, and Satisfiability and Their Applications. *Mathematics in Computer Science*, 11(3-4):483–502, 2017.

[Stu18] Thomas Sturm. Thirty Years of Virtual Substitution: Foundations, Techniques, Applications. In *Symbolic and Algebraic Computation (ISSAC'18)*, pages 11–16, 2018.

[Sur59] János Surányi. *Reduktionstheorie des Entscheidungsproblems im Prädikatenkalkül der ersten Stufe*. Verlag der Ungarischen Akademie der Wissenschaften, 1959.

[Sut18] Geoff Sutcliffe. The 9th IJCAR Automated Theorem Proving System Competition – CASC-J9. *AI Communications*, 31(6):495–507, 2018.

[SV06] Helmut Seidl and Kumar Neeraj Verma. Cryptographic Protocol Verification Using Tractable Classes of Horn Clauses. In *Program Analysis and Compilation, Theory and Practice, Essays Dedicated to Reinhard Wilhelm on the Occasion of His 60th Birthday*, pages 97–119, 2006.

[SV08] Helmut Seidl and Kumar Neeraj Verma. Flat and one-variable clauses: Complexity of verifying cryptographic protocols with single blind copying. *ACM Transactions on Computational Logic*, 9(4):28:1–28:45, 2008.

[SVW16] Thomas Sturm, Marco Voigt, and Christoph Weidenbach. Deciding First-Order Satisfiability when Universal and Existential Variables are Separated. In *Logic in Computer Science (LICS'16)*, pages 86–95. IEEE/ACM, 2016. An extended version is available at the arXiv preprint server (`arXiv.org`) under the signature arXiv:1511.08999 [cs.LO].

[SWW10] Martin Suda, Christoph Weidenbach, and Patrick Wischnewski. On the Saturation of YAGO. In *Automated Reasoning (IJCAR'10)*, pages 441–456, 2010.

[Sze88] Róbert Szelepcsényi. The Method of Forced Enumeration for Nondeterministic Automata. *Acta Informatica*, 26(3):279–284, 1988.

[Tam91] Tanel Tammet. *Resolution methods for decision problems and finite-model building*. PhD thesis, Chalmers University of Technology, Göteborg, Sweden, 1991.

[Tam95] Tanel Tammet. Using Resolution for Extending KL-ONE-type Languages. In *Information and Knowledge Management (CIKM'95)*, pages 326–332, 1995.

[Tar54] Alfred Tarski. Contributions to the theory of models. I. *Indagationes Mathematicae*, XVI:572–581, 1954.

[Tar57] Alfred Tarski. A Decision Method for Elementary Algebra and Geometry. Technical Report R-109, RAND Corporation, 1948. Revised in 1951. Second Edition 1957. Prepared for publication by J.C.C. McKinsey. Reprinted in 1998 [Tar98].

[Tar98] Alfred Tarski. A Decision Method for Elementary Algebra and Geometry. In Bob F. Caviness and Jeremy R. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, pages 24–84. Springer, 1998.

[Teu17] Andreas Teucke. *An Approximation and Refinement Approach to First-Order Automated Reasoning*. PhD thesis, Department of Computer Science, Saarland University, 2017.

[TH96] Cesare Tinelli and Mehdi T. Harandi. A New Correctness Proof of the Nelson–Oppen Combination Procedure. In *Frontiers of Combining Systems (FroCoS'96)*, pages 103–119, 1996.

[TR03] Cesare Tinelli and Christophe Ringeissen. Unions of non-disjoint theories and combinations of satisfiability procedures. *Theoretical Computer Science*, 290(1):291–353, 2003.

[Tra84] Boris A. Trakhtenbrot. A Survey of Russian Approaches to Perebor (Brute-Force Search) Algorithms. *IEEE Annals of the History of Computing*, 6(4):384–400, 1984.

[TRRK10] Duc-Khanh Tran, Christophe Ringeissen, Silvio Ranise, and Hélène Kirchner. Combination of convex theories: Modularity, deduction completeness, and explanation. *Journal of Symbolic Computation*, 45(2):261–286, 2010.

[TS96] A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*, volume 43 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1996.

[TSSP04] Muralidhar Talupur, Nishant Sinha, Ofer Strichman, and Amir Pnueli. Range Allocation for Separation Logic. In *Computer Aided Verification (CAV'04)*, pages 148–161, 2004.

[Tur36] Alan Mathison Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265, 1936.

[Tur38] Alan Mathison Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. A Correction. *Proceedings of the London Mathematical Society*, s2-43(1):544–546, 1938.

[TW15] Andreas Teucke and Christoph Weidenbach. First-Order Logic Theorem Proving and Model Building via Approximation and Instantiation. In *Frontiers of Combining Systems (FroCoS'15)*, pages 85–100, 2015.

[TW17] Andreas Teucke and Christoph Weidenbach. Decidability of the Monadic Shallow Linear First-Order Fragment with Straight Dismatching Constraints. In *Automated Deduction (CADE-26)*, pages 202–219, 2017.

[TZ05] Cesare Tinelli and Calogero G. Zarba. Combining Nonstably Infinite Theories. *Journal of Automated Reasoning*, 34(3):209–238, 2005.

[Vää07] Jouko A. Väänänen. *Dependence Logic – A New Approach to Independence Friendly Logic*, volume 70 of *London Mathematical Society student texts*. Cambridge University Press, 2007.

[vB97] Johan van Benthem. Dynamic bits and pieces. Technical Report LP-97-01, Institute for Logic, Language and Computation (ILLC), University of Amsterdam, 1997.

[vD13] Dirk van Dalen. *Logic and Structure*. Springer, fifth edition, 2013.

[Vea97a] Margus Veanes. Computational Complexity of Basic Decision Problems of Finite Tree Automata. Technical Report UPMAIL 133, Computing Science Department, Uppsala University, 1997.

[Vea97b] Margus Veanes. *On Simultaneous Rigid E-Unification*. PhD thesis, Computing Science Department, Uppsala University, 1997.

[vH02] Jean van Heijenoort. *From Frege to Gödel – A Source Book in Mathematical Logic, 1879–1931*. Harvard University Press, 2002.

[Voi17a] Marco Voigt. The Bernays–Schönfinkel–Ramsey Fragment with Bounded Difference Constraints over the Reals is Decidable. In *Frontiers of Combining Systems (FroCoS'17)*, pages 244–261, 2017. An extended version is available at the arXiv preprint server (`arXiv.org`) under the signature arXiv:1706.08504 [cs.LO].

[Voi17b] Marco Voigt. A Fine-Grained Hierarchy of Hard Problems in the Separated Fragment. In *Logic in Computer Science (LICS'17)*, pages 1–12. IEEE/ACM, 2017. An extended version is available at the arXiv preprint server (`arXiv.org`) under the signature arXiv:1704.02145 [cs.LO].

[Voi17c] Marco Voigt. On Generalizing Decidable Standard Prefix Classes of First-Order Logic. *ArXiv preprint*, (arXiv:1706.03949 [cs.LO]), 2017.

[Voi17d] Marco Voigt. Towards Elimination of Second-Order Quantifiers in the Separated Fragment. In *Proceedings of the Workshop on Second-Order Quantifier Elimination and Related Topics (SOQE 2017)*, pages 67–81, 2017.

[VW15] Marco Voigt and Christoph Weidenbach. Bernays–Schönfinkel–Ramsey with Simple Bounds is NEXPTIME-complete. *ArXiv preprint*, arXiv:1501.07209 [cs.LO], 2015.

[Wei88] Volker Weispfenning. The Complexity of Linear Problems in Fields. *Journal of Symbolic Computation*, 5(1/2):3–27, 1988.

[Wei97] Volker Weispfenning. Quantifier Elimination for Real Algebra – the Quadratic Case and Beyond. *Applicable Algebra in Engineering, Communication and Computing*, 8(2):85–101, 1997.

[Wei98] Christoph Weidenbach. Sorted Unification and Tree Automata. In Wolfgang Bibel and Peter H. Schmitt, editors, *Automated Deduction – A Basis for Applications. Volume I: Foundations – Calculi and Methods*, volume 8 of *Applied Logic Series*, pages 291–320. Kluwer Academic Publishers, 1998.

[Wei99] Christoph Weidenbach. Towards an Automatic Analysis of Security Protocols in First-Order Logic. In *Automated Deduction (CADE-16)*, LNCS 1632, pages 314–328. Springer, 1999.

[Wer15a] Christoph Wernhard. Heinrich Behmann's Contributions to Second-Order Quantifier Elimination from the View of Computational Logic. Technical report, TU Dresden, 2015.

[Wer15b] Christoph Wernhard. Second-Order Quantifier Elimination on Relational Monadic Formulas – A Basic Method and Some Less Expected Applications. In *Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX'15)*, pages 253–269, 2015.

[WGG06] Chao Wang, Aarti Gupta, and Malay K. Ganai. Predicate Learning and Selective Theory Deduction for a Difference Logic Solver. In *Design Automation Conference (DAC'06)*, pages 235–240, 2006.

[Wil86] H. Paul Williams. Fourier's Method of Linear Programming and Its Dual. *The American Mathematical Monthly*, 93(9):681–695, 1986.

[Wir76] Martin Wirsing. *Das Entscheidungsproblem der Prädikatenlogik 1. Stufe mit Identität und Funktionszeichen in Herbrandformeln.* PhD thesis, Ludwig–Maximilians–Universität München, 1976.

[Wir77] Martin Wirsing. Das Entscheidungsproblem der Klasse von Formeln, die höchstens zwei Primformeln enthalten. *Manuscripta Mathematica*, 22:13–25, 1977.

[Wir78] Martin Wirsing. Kleine unentscheidbare Klassen der Prädikatenlogik mit Identität und Funktionszeichen. *Archiv für mathematische Logik und Grundlagenforschung*, 19(1):97–109, 1978.

[Wis12] Patrick Wischnewski. *Efficient Reasoning Procedures for Complex First-Order Theories.* PhD thesis, Department of Computer Science, Saarland University, 2012.

[WPK09] Thomas Wies, Ruzica Piskac, and Viktor Kuncak. Combining Theories with Shared Set Operations. In *Frontiers of Combining Systems (FroCoS'09)*, LNCS 5749, pages 366–382. Springer, 2009.

[Zam87] N. K. Zamov. On a Connection Between the Resolution Method and the Inverse Method. In *Fundamentals of Computation Theory (FCT'87)*, pages 501–505, 1987.

Index

2SAT, 143
3SAT, 145
$\exists^*$-GBSR, 140
$\exists^*$-SF, 140
$\mathcal{A}$-color, 235
$\mathcal{A}$-coloring, 235
$\mathcal{J}_{\mathcal{A}}$-equivalence, 234
$\mathcal{J}_{\mathcal{A}}$-uniformity, 220, 235
$\widehat{\simeq}_\kappa$-uniformity, 220
$\simeq_\kappa$-uniformity, 220
Π^0_1, 277
Σ^0_1, 277

abstraction, 224
Ackermann fragment (AF), 3, **24**, 42, 67, 70, 76, 77, 80, 174, 222, 293–295
additive constraint, 218, 265
almost uninterpreted formulas, 288
analytical hierarchy, 278
anti-prenexing, 191
argument position, 103
arithmetical hierarchy, 276
array property fragment, 219, 298–301
atom
  Σ-$\sim$, 9
  linear $\sim$, 89

Basic Elimination Lemma, 208
Bell number, 148
benign co-occurrence, 55, 56, 58
Bernays–Schönfinkel fragment (BS), **24**, 147, 168, 298, 299
Bernays–Schönfinkel–Ramsey fragment (BSR), 3, 5, 19, **24**, 29, 39, 61, 102, 104, 124, 157, 218, 222, 227, 240, 243, 291, 293–296, 299
block-separated fragment of linear rational arithmetic (BSF-LRA), 190
Boolean combination, **9**, 26, 96, 100, 254
BSR with bounded difference constraints (BSR(BD)), 5, 219, **227**, 245, 296–301
BSR with simple linear rational constraints (BSR(SLR)), 5, 219, **227**, 233, 242, 251, 296–301
BSR(BD) normal form, 228
BSR(SLR) normal form, 228
BSR-Sat, 140, 241, 253

Church–Turing thesis, 1
classical decision problem, **23**, 139, 181, 291
clause, 10
  Ackermann-like $\sim$, 177
  Bernays–Schönfinkel–Ramsey $\sim$ (BSR clause), 224
  unit $\sim$, 10
clause set
  encoding length of a $\sim$, 224
  length of a $\sim$, 224
  saturated $\sim$, 170, 176
clique-guarded fragment, 27
clock, 254
  atomic $\sim$ constraint, 254
  constraint, 254
  valuation, 255
  variable, 254
coloring, 231
  $\mathcal{A}$-$\sim$, 235
compactness, 116, 276
companion
  conjunctive $\sim$, 63
  disjunctive $\sim$, 63
completeness
  model $\sim$, 280
  refutational $\sim$, 170, 276, 280
  sufficient $\sim$, 276
configuration of a two-counter machine, 265
conjunctive companion, 63
conjunctive normal form (CNF), 10, 28, 62, 145, 276
connected component, 56
constraint satisfaction problem (CSP), 191
counting quantifier, 30, 40
Craig–Lyndon interpolation, 6, 167, 168, 174, 293

decision procedure, 20, 23, 147, 204, 219, 221, 229, 235, 238, 240, 242, 251, 253, 286, 295, 297, 301
degree of interaction
  for GAF sentences, 137
  for GBSR sentences, **58**, 66, 139
  of existential variables, **31**, 139
  of universal variables, **122**, 139
dependence
  finitely controllable $\sim$, **112**, 124
  strong $\sim$, 19, 20, 28, **111**, 124
  weak $\sim$, 5, 6, 19, 20, 28, **111**, 122, 124, 293, 295
dependence logic, 192, 295
dependency scheme, 191
diagonal constraint, 255
difference constraint, 5, 218, 219, 221, **227**, 254, 265, 279, 284, 297
  bounded $\sim$, 227
difference constraint graph, 228
difference logic, 228, 254
disjunctive companion, 63
disjunctive normal form (DNF), 10, 62
distance
  of domain elements, 46
domain, 11
  Herbrand $\sim$, **13**, 103
domino problem, 3
  bounded $\sim$, 45, 149, **150**
  unconstrained $\sim$, **40**, 150
domino system
  bounded $\sim$, 149
  unconstrained $\sim$, 41

effectively propositional logic (EPR), 294
ELEMENTARY, 140
elimination lemma, 208
elimination set, 21, 186, 229
entailment
  semantic $\sim$, 12, 224
equality over uninterpreted functions (EUF)
  theory of $\sim$, 242, **285**, 288, 298
equisatisfiability, 11
equivalence
  $\mathcal{T}$-$\sim$, 12
  for timed automata, 255
  semantic $\sim$, 12, 224
equivalence class, 14
essentially uninterpreted fragment, 301
existential first-order fragment ($\exists$FO), 42, 140, **142**
existential rule, 28
exponential-time hierarchy, 142

fingerprint, 112, 114, **117**, 125, 181, 187, 196, 197
fingerprint function, 114, 117, **118**, **125**, 181, **189**, 197
  $\lambda_{x,\ell}$, 125
  μ_ℓ, 189
  $\mu_{\ell,k}$, 118
finite model property, **23**, 24, 25, 29, 36, 61, 103, 109, 112, 122, 124, 139, 220, 296
finite-variable logic, 92, **95**, 141
fluted fragment (FL), 3, **26**, 39, 42, 99, 100, 115, 141
FOL(LA) encoding, 256
fomula
  Horn, 275, 297
  Krom, 275, 297
formula
  Σ-$\sim$, 9
  r-local $\sim$, 46, 55
  atomic Σ-$\sim$, 9
  basic $\sim$, 18, 94, 96, 101, 192
  binary encoding of a $\sim$, 11
  closed $\sim$, 10
  encoding length of a $\sim$, 11, 224
  first-order $\sim$, 9
  GF $\sim$, 83
  GNFO $\sim$, 93
  ground $\sim$, 10
  guarded-negation $\sim$, 92
  guarded $\sim$, 83
  Horn $\sim$, **10**, 27, 37, 39, 104, 106, 143, 146, 147, 154, 156, 157, 266, 276, 279
  in conjunctive normal form, 10
  in disjunctive normal form, 10
  in negation normal form, 10
  in prenex normal form, 10
  in standard form, 10
  Krom $\sim$, **10**, 25, 28, 37, 39, 143, 146, 147
  length of a $\sim$, 10
  LGF $\sim$, 83
  local $\sim$, 46
  loosely guarded $\sim$, 83
  LRA+PN $\sim$, 224
  LRA $\sim$, 182, 223
  matrix of a $\sim$, 10
  monadic $\sim$, 22
  PA+P $\sim$, 223
  PA $\sim$, 223
  positive $\sim$, 185
  propositional $\sim$, 143
  quantified Σ-$\sim$, 9
  relational $\sim$, 10
  satisfied $\sim$, 11
  second-order $\sim$, 14

segregated ∼, 19
separated guarded-negation ∼, 93
separated loosely guarded ∼, 83
SGF ∼, 84
SGNFO ∼, 93
SLGF ∼, 84
Fourier–Motzkin elimination, 20, 230
fractional part, 225

GAF special form, 70, 74
GAF-Sat, 70
Gaifman graph, 46
Gaifman normal form, 46, 61, 292
GBSR with simple linear rational constraints (GBSR(SLR)), 245
GBSR-Sat, 57, 61, 140
generalized Ackermann fragment (GAF), 3, **67**, 102, 292, 293, 295
generalized Bernays–Schönfinkel–Ramsey fragment (GBSR), 3, **56**, 102–104, 139, 220, 244, 291–293, 295, 298
generalized Gödel–Kalmár–Schütte fragment (GGKS), 3, **77**, 102, 292, 293
GGKS special form, 80
GGKS-Sat, 80
GNFO-Sat, 93
Gödel–Kalmár–Schütte fragment (GKS), 3, **24**, 42, 77, 293
guard, 21, 83, 168, 295
atomic ∼, 83, 92
loose ∼, 83
negation ∼, 92
separated negation ∼, 93
guarded fragment (GF), 3, **26**, 39, **83**, 168, 222, 291, 294
guarded-negation fragment (GNFO), 3, **27**, **93**, 292
Gurevich–Maslov–Orevkov fragment, **24**, 67, 76

H_1, 27, 106
halting problem, 265, 297
hashtable property fragment, 219
Henkin quantifier, 192, 295
Herbrand domain, **13**, 103, 107
Herbrand fragment, 28
Herbrand model
least ∼, **13**, 27
minimal ∼, **13**, 27, 105, 106
Herbrand structure, **13**, 103, 105, 106, 143
Herzig's ordered fragment, **26**, 99, 100
hierarchic superposition, 242, 276, 301

independence logic, 192, 295
infinity axiom, 24, 40
integral part, 225
interpolant, 167, 168, 174
interpolation
Craig–Lyndon ∼, 6, 167, 168, 174, 293
intersection non-emptiness problem, 104
inverse method, 25

Lewis' fragment T, 27
lexicographic path ordering (LPO), 169
linear program, 241
linear rational arithmetic (LRA), 6, 20, 181, **182**, 217, **223**, 297
linear-time temporal logic, 222
literal, 10
$LK_{\top\bot}$, 174
Löb–Gurevich fragment, 24, 31, 76, 102
location, 254
initial ∼, 254
location invariant, 255
location transition relation, 255
loosely guarded fragment (LGF), 3, **26**, **83**, 168, 222, 291, 294
Löwenheim fragment, 23
LRA formula, 223
LRA term, 223
LRA+PN formula, 224
LRA+PN term, 224

Maslov fragment, **25**, 28, 147, 149
Maslov's fragment K, **25**, 42, 56, 76, 80, 99, 142, 291, 295
matrix, 10
metric temporal logic, 222
miniscoping, 191
Minsky machine, 1, 265
model, 11
model-checking game, 5, 20, 57, 70, **113**, 117, 124, 197
monadic first-order fragment (MFO), 3, 22, **23**, 29, 36, 39, 70, 76, 80, 84, 93, 95, 100, 102, 104, 115, 123, 141, 147, 157, 204, 222, 291, 292, 294, 298
monadic second-order fragment (MSO), 23, 29, 204, 212
monadic second-order theory of one successor (S1S), 220
monadic shallow linear Horn fragment (MSLH), **27**, 104
multiplicative constraint, 218, 265

negation
guarded ∼, 92
scope of a ∼ sign, 9
negation normal form, 10

negative occurrence, 168
neighborhood
  r-$\sim$, 46
Nelson–Oppen combination framework, 221, **242**, 285, 294, 298

one-free fragment, 28
outcome, **117**, 125

PA formula, 223
PA term, 223
PA+P formula, 223
PA+P term, 223
packed guarded fragment, 27
path, 228
  length of a $\sim$, 228
  simple $\sim$, 228
polarity, **21**, 22, 156
polynomial-time hierarchy (PH), 142
positive occurrence, 168
positive variable dominated clause fragment (PVD), 28
power set, 14
precedence, 169
prenex normal form, 10
preorder, 239
Presburger arithmetic, 4, 6, 20, 217, 223, 263, 264, 297
  universal fragment of $\sim$, 264
pseudo-integers, 281
purification, 224

quantified Boolean formula (QBF), 142, 191
quantifier
  block, 9
  counting $\sim$, 30
  guarded $\sim$, 83
  Henkin $\sim$, 192
  leading $\sim$, 29
  rank, 10
  scope of a $\sim$, 9, 83, 193
quantifier elimination, **20**, 299
  first-order $\sim$, 20, 183, 185, 217, 244
  second-order $\sim$, 22, 29, 204
quantifier shifting, **12**, 15, 35, 56, 62, 67
quotient constraint, 218, 265
quotient set, 14

Rabin fragment, 24, 28, 76
Ramsey theory, 24, 231
reachability problem, 6, 222, 298
recurrence problem, 275, 297
reference variable, 67, 78, 113, 124
refinement of an equivalence relation, 14
refutational completeness, 170, 276, 280
representative, 107, 119, 127, 198
resolution
  ordered $\sim$, 169
rewrite function, 148
run
  of a two-counter machine, 265

SAT, 143, 147
  Horn-$\sim$, 143, 147
  Krom-$\sim$, 143, 147
satisfiability, 11
saturation, 170
scope
  of a negation sign, 9
  of a quantifier, 9, 83, 193
Scott normal form, 97
selection function, **33**, 205
  for ordered resolution, 170
sentence
  Σ-$\sim$, 10
  basic local $\sim$, 46
  inconsistent $\sim$, 11
  invalid $\sim$, 11
  monadic $\sim$, 22
  satisfiable $\sim$, 11
  valid $\sim$, 11
separated k-variable fragment (SFO^k), 95
separated fluted fragment (SFL), 3, 99, **100**, 102, 292
separated fragment (SF), 3, 19, **29**, 102, 104, 124, 139, 220, 244, 291–293, 295, 298
separated guarded fragment (SGF), 3, **83**, 102, 291, 292, 295
separated guarded-negation fragment (SGNFO), 3, **93**, 102, 292
separated loosely guarded fragment (SLGF), 3, **83**, 102, 291, 292, 295
separated two-variable fragment (SFO^2), 3, 96, 102, 292, 293
separateness of variables, 15
  strict $\sim$, 15
separation logic, 286, 298
separation predicate, 219
sequent calculus, 174
SF-Sat, 29, 31, 40, 45, 139
SFL-Sat, 101
SFO^2-Sat, 96
SGF-Sat, 84
SGNFO-Sat, 95
Shelah fragment, **24**, 67, 76
Shostak combination framework, 242
signature, 9
simple cycle, 228

Skolem constant, 12, 224
Skolem fragment, 25, **25**, 42, 291
Skolem function, 12, 187
Skolem term, 12
Skolemization, 6, **12**, 19, 111, 191, 293, 295
  exhaustive ∼, 12
SLGF-Sat, 84
small model property, **23**, 36, 61, 122, 124, 139
sort
  background ∼, 223
  base ∼, 223
  free ∼, 223
  uninterpreted ∼, 223
spectrum, 294
stable infiniteness, 243
standard form, 10
standard translation, 26
state transition system, 255
strategy, 5, 20, 111, 113, **117**, 124, **125**, 197
  λ-semi-uniform ∼, 125, **126**
  μ-uniform ∼, 117, **118**, 124, 190
  ν-ξ-uniform ∼, **198**, 202
  satisfying ∼, 5, 20, 113, 115, **117**, 124, **125**, 190, 196
  semi-uniform ∼, 113, 125
  target set of a ∼, 123, 130, 136, 190
  uniform ∼, 113, 115, 117
  winning ∼, 5, 38, 113, 117
stratified occurrences of function symbols, 103
stratified vocabulary, **103**, 299
strongly separated fragment (SSF), **32**, 36, 42, 141, 190
structure
  Σ-∼, **11**
  $\widehat{\simeq}_\kappa$-uniform ∼, 246
  $\simeq_\kappa$-uniform ∼, 246
  Herbrand ∼, **13**, 103, 105, 143
  sub-∼, 12
  uniform ∼, 220, 299
substitution, 10, 170
  sequential ∼, 10
  simultaneous ∼, 10
  virtual ∼, 20, **185**, 229, 244
Substructure Lemma, 12
sufficient completeness, 276
superposition
  hierarchic, 222, 242, 276, 277, 280, 298, 301

TA region, 254, **256**
target set, 123, 130, 136, 190
term
  Σ-∼, 9
  encoding length of a ∼, 11
  evaluation of a ∼, 11
  formal ∼, 186
  ground ∼, 10, 219, 227
  length of a ∼, 10
  linear ∼, 27, 106
  LRA+PN ∼, 224
  LRA ∼, 223
  PA+P ∼, 223
  PA ∼, 223
  shallow ∼, 27
  Skolem ∼, 12
  unifiable ∼s, 170
testpoint, 21, 185, 229
tetration, 14
theory
  background ∼, 223
  base ∼, 223
  decidable Σ-∼, 217
  logical Σ-∼, 12
  stably infinite ∼, 243
tiling, 41
time constructible function, 150
timed automaton, 6, 222, 254, **254**, 298
Tower, 140
transition guard, 255
tree, 46
tree automaton, 104, 105
tree-like model property, 27
tuple
  ascending, 231
Turing machine, 1, 150, 265
  recurring ∼, 263
  simple ∼, 150
two-counter machine, **265**, 297
  nondeterministic ∼, 278
  recurring ∼, 275, 278, 297
two-variable fragment (FO^2), 3, **25**, 39, 42, 95, 96, 222, 293
type, 187

ultimately periodic set, 299
unary-negation fragment, 28, 291
unifiable terms, 170
unifier, 170
  most general ∼ (mgu), 170
uniform one-dimensional fragment, 28, 291
uniformly colored set, 231
uninterpreted function symbol, 223
uninterpreted predicate symbol, 223
uninterpreteed sort, 223
unit clause, 10
universe, 11

upward closure, 69, 78, 193
    extended ∼, 193

variable
    block index of a ∼, 193
    bound first-order ∼, 10
    bound second-order ∼, 14
    clock ∼, 254
    first-order ∼, 9
    free first-order ∼, 10
    free second-order ∼, 14
    guard-separated sets of ∼s, 83
    index of a ∼, 32, 58, 67
    interaction of ∼s, 31
    reference ∼, 67, 78, 113, 124
    second-order ∼, 14, 22
    separated sets of ∼s, 15
    strictly separated sets of ∼s, 15
variable assignment, 11
    explicit definition of a ∼, 11
    update of a ∼, 11
virtual substitution, 20, **185**, 229, 244
vocabulary, 9
    relational ∼, 9
    stratified ∼, 102, **103**, 299, 301