Wolfgang Rautenberg

A Concise Introduction to Mathematical Logic

Textbook
Third Edition

Typeset and layout: The author

Version from June 2009

corrections included

Foreword

by Lev Beklemishev, Moscow

The field of mathematical logic—evolving around the notions of logical validity, provability, and computation—was created in the first half of the previous century by a cohort of brilliant mathematicians and philosophers such as Frege, Hilbert, Gödel, Turing, Tarski, Malcev, Gentzen, and some others. The development of this discipline is arguably among the highest achievements of science in the twentieth century: it expanded mathematics into a novel area of applications, subjected logical reasoning and computability to rigorous analysis, and eventually led to the creation of computers.

The textbook by Professor Wolfgang Rautenberg is a well-written introduction to this beautiful and coherent subject. It contains classical material such as logical calculi, beginnings of model theory, and Gödel's incompleteness theorems, as well as some topics motivated by applications, such as a chapter on logic programming. The author has taken great care to make the exposition readable and concise; each section is accompanied by a good selection of exercises.

A special word of praise is due for the author's presentation of Gödel's second incompleteness theorem, in which the author has succeeded in giving an accurate and simple proof of the derivability conditions and the provable Σ_1-completeness, a technically difficult point that is usually omitted in textbooks of comparable level. This work can be recommended to all students who want to learn the foundations of mathematical logic.

Preface

The third edition differs from the second mainly in that parts of the text have been elaborated upon in more detail. Moreover, some new sections have been added, for instance a separate section on Horn formulas in Chapter 4, particularly interesting for logic programming. The book is aimed at students of mathematics, computer science, and linguistics. It may also be of interest to students of philosophy (with an adequate mathematical background) because of the epistemological applications of Gödel's incompleteness theorems, which are discussed in detail.

Although the book is primarily designed to accompany lectures on a graduate level, most of the first three chapters are also readable by undergraduates. The first hundred twenty pages cover sufficient material for an undergraduate course on mathematical logic, combined with a due portion of set theory. Only that part of set theory is included that is closely related to mathematical logic. Some sections of Chapter 3 are partly descriptive, providing a perspective on decision problems, on automated theorem proving, and on nonstandard models.

Using this book for independent and individual study depends less on the reader's mathematical background than on his (or her) ambition to master the technical details. Suitable examples accompany the theorems and new notions throughout. We always try to portray simple things simply and concisely and to avoid excessive notation, which could divert the reader's mind from the essentials. Line breaks in formulas have been avoided. To aid the student, the indexes have been prepared very carefully. Solution hints to most exercises are provided in an extra file ready for download from Springer's or the author's website.

Starting from Chapter 4, the demands on the reader begin to grow. The challenge can best be met by attempting to solve the exercises without recourse to the hints. The density of information in the text is rather high; a newcomer may need one hour for one page. Make sure to have paper and pencil at hand when reading the text. Apart from sufficient training in logical (or mathematical) deduction, additional prerequisites are assumed only for parts of Chapter 5, namely some knowledge of classical algebra, and at the very end of the last chapter some acquaintance with models of axiomatic set theory.

On top of the material for a one-semester lecture course on mathematical logic, basic material for a course in logic for computer scientists is included in Chapter **4** on logic programming. An effort has been made to capture some of the interesting aspects of this discipline's logical foundations. The resolution theorem is proved constructively. Since all recursive functions are computable in PROLOG, it is not hard to deduce the undecidability of the existence problem for successful resolutions.

Chapter **5** concerns applications of mathematical logic in mathematics itself. It presents various methods of model construction and contains the basic material for an introductory course on model theory. It contains in particular a model-theoretic proof of quantifier eliminability in the theory of real closed fields, which has a broad range of applications.

A special aspect of the book is the thorough treatment of Gödel's incompleteness theorems in Chapters **6** and **7**. Chapters **4** and **5** are not needed here. **6.1**[1] starts with basic recursion theory needed for the arithmetization of syntax in **6.2** as well as in solving questions about decidability and undecidability in **6.5**. Defining formulas for arithmetical predicates are classified early, to elucidate the close relationship between logic and recursion theory. Along these lines, in **6.5** we obtain in one sweep Gödel's first incompleteness theorem, the undecidability of the tautology problem by Church, and Tarski's result on the nondefinability of truth, all of which are based on certain diagonalization arguments. **6.6** includes among other things a sketch of the solution to Hilbert's tenth problem.

Chapter **7** is devoted mainly to Gödel's second incompleteness theorem and some of its generalizations. Of particular interest thereby is the fact that questions about self-referential arithmetical statements are algorithmically decidable due to Solovay's completeness theorem. Here and elsewhere, Peano arithmetic (PA) plays a key role, a basic theory for the foundations of mathematics and computer science, introduced already in **3.3**. The chapter includes some of the latest results in the area of self-reference not yet covered by other textbooks.

Remarks in small print refer occasionally to notions that are undefined and direct the reader to the bibliography, or will be introduced later. The bibliography can represent an incomplete selection only. It lists most

[1] This is to mean Section **6.1**, more precisely, Section **1** in Chapter **6**. All other boldface labels are to be read accordingly throughout the book.

English textbooks on mathematical logic and, in addition, some original papers mainly for historical reasons. It also contains some titles treating biographical, historical, and philosophical aspects of mathematical logic in more detail than this can be done in the limited size of our book. Some brief historical remarks are also made in the *Introduction.* Bibliographical entries are sorted alphabetically by author names. This order may slightly diverge from the alphabetic order of their citation labels.

The material contained in this book will remain with high probability the subject of lectures on mathematical logic in the future. Its streamlined presentation has allowed us to cover many different topics. Nonetheless, the book provides only a selection of results and can at most accentuate certain topics. This concerns above all Chapters **4**, **5**, **6**, and **7**, which go a step beyond the elementary. Philosophical and foundational problems of mathematics are not systematically discussed within the constraints of this book, but are to some extent considered when appropriate.

The seven chapters of the book consist of numbered sections. A reference like Theorem 5.4 is to mean Theorem 4 in Section **5** of a given chapter. In cross-referencing from another chapter, the chapter number will be adjoined. For instance, Theorem 6.5.4 means Theorem 5.4 in Chapter **6**. You may find additional information about the book or contact me on my website `www.math.fu-berlin.de/~raut`. Please contact me if you propose improved solutions to the exercises, which may afterward be included in the separate file *Solution Hints to the Exercises*.

I would like to thank the colleagues who offered me helpful criticism along the way. Useful for Chapter **7** were hints from Lev Beklemishev and Wilfried Buchholz. Thanks also to Peter Agricola for his help in parts of the contents and in technical matters, and to Michael Knoop, Emidio Barsanti, and David Kramer for their thorough reading of the manuscript and finding a number of mistakes.

Wolfgang Rautenberg, June 2009

Contents

Introduction

Traditional logic as a part of philosophy is one of the oldest scientific disciplines. It can be traced back to the Stoics and to Aristotle[1] and is the root of what is nowadays called philosophical logic. Mathematical logic, however, is a relatively young discipline, having arisen from the endeavors of Peano, Frege, and Russell to reduce mathematics entirely to logic. It steadily developed during the twentieth century into a broad discipline with several subareas and numerous applications in mathematics, computer science, linguistics, and philosophy.

One feature of modern logic is a clear distinction between object language and metalanguage. The first is formalized or at least formalizable. The latter is, like the language of this book, a kind of a colloquial language that differs from author to author and depends also on the audience the author has in mind. It is mixed up with semiformal elements, most of which have their origin in set theory. The amount of set theory involved depends on one's objectives. Traditional semantics and model theory as essential parts of mathematical logic use stronger set-theoretic tools than does proof theory. In some model-theoretic investigations these are often the strongest possible ones. But on average, little more is assumed than knowledge of the most common set-theoretic terminology, presented in almost every mathematical course or textbook for beginners. Much of it is used only as a *façon de parler.*

The language of this book is similar to that common to almost all mathematical disciplines. There is one essential difference though. In mathematics, metalanguage and object language strongly interact with each other, and the latter is semiformalized in the best of cases. This method has proved successful. Separating object language and metalanguage is relevant only in special context, for example in axiomatic set theory, where formalization is needed to specify what certain axioms look like. Strictly formal languages are met more often in computer science. In analyzing complex software or a programming language, as in logic, formal linguistic entities are the central objects of consideration.

[1] The Aristotelian syllogisms are easy but useful examples for inferences in a first-order language with unary predicate symbols. One of these syllogisms serves as an example in Section **4.6** on logic programming.

The way of arguing about formal languages and theories is traditionally called the *metatheory*. An important task of a metatheoretic analysis is to specify procedures of logical inference by so-called *logical calculi*, which operate purely syntactically. There are many different logical calculi. The choice may depend on the formalized language, on the logical basis, and on certain aims of the formalization. Basic metatheoretic tools are in any case the naive natural numbers and inductive proof procedures. We will sometimes call them proofs by *metainduction*, in particular when talking about formalized object theories that speak about natural numbers. Induction can likewise be carried out on certain sets of strings over a fixed alphabet, or on the system of rules of a logical calculus.

The logical means of the metatheory are sometimes allowed or even explicitly required to be different from those of the object language. But in this book the logic of object languages, as well as that of the metalanguage, are classical, two-valued logic. There are good reasons to argue that classical logic is the logic of common sense. Mathematicians, computer scientists, linguists, philosophers, physicists, and others are using it as a common platform for communication.

It should be noticed that logic used in the sciences differs essentially from logic used in everyday language, where logic is more an art than a serious task of saying what follows from what. In everyday life, nearly every utterance depends on the context. In most cases logical relations are only alluded to and rarely explicitly expressed. Some basic assumptions of two-valued logic mostly fail, in particular, a context-free use of the logical connectives. Problems of this type are not dealt with here. To some extent, many-valued logic or Kripke semantics can help to clarify the situation, and sometimes intrinsic mathematical methods must be used in order to solve such problems. We shall use Kripke semantics here for a different goal, though, the analysis of self-referential sentences in Chapter **7**.

Let us add some historical remarks, which, of course, a newcomer may find easier to understand *after* and not *before* reading at least parts of this book. In the relatively short period of development of modern mathematical logic in the twentieth century, some highlights may be distinguished, of which we mention just a few. Many details on this development can be found in the excellent biographies [Daw] and [FF] on Gödel and Tarski, the leading logicians in the last century.

The first was the axiomatization of set theory in various ways. The most important approaches are those of Zermelo (improved by Fraenkel and von Neumann) and the theory of types by Whitehead and Russell. The latter was to become the sole remnant of Frege's attempt to reduce mathematics to logic. Instead it turned out that mathematics can be based entirely on set theory as a first-order theory. Actually, this became more salient after the rest of the hidden assumptions by Russell and others were removed from axiomatic set theory around 1915; see [Hei]. For instance, the notion of an ordered pair, crucial for reducing the notion of a function to set theory, is indeed a set-theoretic and not a logical one.

Right after these axiomatizations were completed, Skolem discovered that there are countable models of the set-theoretic axioms, a drawback to the hope for an axiomatic characterization of a set. Just then, two distinguished mathematicians, Hilbert and Brouwer, entered the scene and started their famous quarrel on the foundations of mathematics. It is described in a comprehensive manner for instance in [Kl2, Chapter IV] and need therefore not be repeated here.

As a next highlight, Gödel proved the completeness of Hilbert's rules for predicate logic, presented in the first modern textbook on mathematical logic, [HA]. Thus, to some extent, a dream of Leibniz became real, namely to create an *ars inveniendi* for mathematical truth. Meanwhile, Hilbert had developed his view on a foundation of mathematics into a program. It aimed at proving the consistency of arithmetic and perhaps the whole of mathematics including its nonfinitistic set-theoretic methods by finitary means. But Gödel showed by his incompleteness theorems in 1931 that Hilbert's original program fails or at least needs thorough revision.

Many logicians consider these theorems to be the top highlights of mathematical logic in the twentieth century. A consequence of these theorems is the existence of consistent extensions of Peano arithmetic in which true and false sentences live in peaceful coexistence with each other, called "dream theories" in **7.3**. It is an intellectual adventure of holistic beauty to see wisdom from number theory known for ages, such as the Chinese remainder theorem, simple properties of prime numbers, and Euclid's characterization of coprimeness (page 249), unexpectedly assuming pivotal positions within the architecture of Gödel's proofs. Gödel's methods were also basic for the creation of recursion theory around 1936.

Church's proof of the undecidability of the tautology problem marks another distinctive achievement. After having collected sufficient evidence by his own investigations and by those of Turing, Kleene, and some others, Church formulated his famous thesis (see **6.1**), although in 1936 no computers in the modern sense existed nor was it foreseeable that computability would ever play the basic role it does today.

Another highlight of mathematical logic has its roots in the work of Tarski, who proved first the undefinability of truth in formalized languages as explained in **6.5**, and soon thereafter started his fundamental work on decision problems in algebra and geometry and on model theory, which ties logic and mathematics closely together. See Chapter **5**.

As already mentioned, Hilbert's program had to be revised. A decisive step was undertaken by Gentzen, considered to be another groundbreaking achievement of mathematical logic and the starting point of contemporary proof theory. The logical calculi in **1.4** and **3.1** are akin to Gentzen's calculi of natural deduction.

We further mention Gödel's discovery that it is not the axiom of choice (AC) that creates the consistency problem in set theory. Set theory with AC and the continuum hypothesis (CH) is consistent, provided set theory without AC and CH is. This is a basic result of mathematical logic that would not have been obtained without the use of strictly formal methods. The same applies to the independence proof of AC and CH from the axioms of set theory by Cohen in 1963.

The above indicates that mathematical logic is closely connected with the aim of giving mathematics a solid foundation. Nonetheless, we confine ourself to logic and its fascinating interaction with mathematics, which characterizes mathematical logic. History shows that it is impossible to establish a programmatic view on the foundations of mathematics that pleases everybody in the mathematical community. Mathematical logic is the right tool for treating the technical problems of the foundations of mathematics, but it cannot solve its epistemological problems.

Notation

We assume that the reader is familiar with the most basic mathematical terminology and notation, in particular with the *union*, *intersection*, and *complementation* of sets, denoted by $\cup$, $\cap$, and $\setminus$, respectively. Here we summarize only some notation that may differ slightly from author to author or is specific for this book. $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ denote the sets of natural numbers including 0, integers, rational, and real numbers, respectively, and $\mathbb{N}_+, \mathbb{Q}_+, \mathbb{R}_+$ the sets of positive members of the corresponding sets. n, m, i, j, k always denote natural numbers unless stated otherwise. Hence, extended notation like $n \in \mathbb{N}$ is mostly omitted.

In the following, M, N denote sets, $M \subseteq N$ denotes inclusion, while $M \subset N$ means proper inclusion (i.e., $M \subseteq N$ and $M \neq N$). As a rule, we write $M \subset N$ only if the circumstance $M \neq N$ has to be emphasized. If M is fixed in a consideration and N varies over subsets of M, then $M \setminus N$ may also be symbolized by $\setminus N$ or $\neg N$.

$\emptyset$ denotes the *empty set*, and $\mathfrak{P}M$ the *power set* ($=$ set of all subsets) of M. If one wants to emphasize that all elements of a set S are sets, S is also called a *system* or *family* of sets. $\bigcup S$ denotes the union of S, that is, the set of elements belonging to at least one $M \in S$, and $\bigcap S$ stands for the intersection of a nonempty system S, the set of elements belonging to all $M \in S$. If $S = \{M_i \mid i \in I\}$ then $\bigcup S$ and $\bigcap S$ are mostly denoted by $\bigcup_{i \in I} M_i$ and $\bigcap_{i \in I} M_i$, respectively.

A *relation* between M and N is a subset of $M \times N$, the set of ordered pairs (a, b) with $a \in M$ and $b \in N$. A precise definition of (a, b) is given on page 114. Such a relation, f say, is said to be a *function* or *mapping from M to N* if for each $a \in M$ there is precisely one $b \in N$ with $(a, b) \in f$. This b is denoted by $f(a)$ or fa or a^f and called the *value of f at a*. We denote a function f from M to N also by $f\colon M \to N$, or by $f\colon x \mapsto t(x)$, provided $f(x) = t(x)$ for some term t (see **2.2**). $\mathit{ran}\, f = \{fx \mid x \in M\}$ is called the *range* of f, and $\mathit{dom}\, f = M$ its *domain*. id_M denotes the *identical function* on M, that is, $\mathrm{id}_M(x) = x$ for all $x \in M$.

$f\colon M \to N$ is *injective* if $fx = fy \Rightarrow x = y$, for all $x, y \in M$, *surjective* if $\mathit{ran}\, f = N$, and *bijective* if f is both injective and surjective. The reader should basically be familiar with this terminology. The phrase "let f be a function from M to N" is sometimes shortened to "let $f\colon M \to N$."

The set of all functions from a set I to a set M is denoted by M^I. If f, g are functions with $ran\, g \subseteq dom\, f$ then $h\colon x \mapsto f(g(x))$ is called their *composition* (or *product*). It will preferably be written as $h = f \circ g$.

Let I and M be sets, $f\colon I \to M$, and call I the *index set*. Then f will often be denoted by $(a_i)_{i \in I}$ and is named, depending on the context, an (indexed) *family*, an *I-tuple*, or a *sequence*. If 0 is identified with $\emptyset$ and $n > 0$ with $\{0, 1, \dots, n-1\}$, as is common in set theory, then M^n can be understood as the set of n-tuples $(a_i)_{i<n} = (a_0, \dots, a_{n-1})$ *of length n* whose members belong to M. In particular, $M^0 = \{\emptyset\}$. Also the set of sequences $(a_1, \dots, a_n)$ with $a_i \in M$ will frequently be denoted by M^n. In concatenating finite sequences, which has an obvious meaning, the *empty sequence* (i.e., $\emptyset$), plays the role of a neutral element. $(a_1, \dots, a_n)$ will mostly be denoted by $\vec{a}$. Note that this is the empty sequence for $n = 0$, similar to $\{a_1, \dots, a_n\}$ for $n = 0$ always being the empty set. $f\vec{a}$ means $f(a_1, \dots, a_n)$ throughout.

If A is an *alphabet*, i.e., if the elements $\mathsf{s} \in \mathsf{A}$ are symbols or at least named symbols, then the sequence $(\mathsf{s}_1, \dots, \mathsf{s}_n) \in \mathsf{A}^n$ is written as $\mathsf{s}_1 \cdots \mathsf{s}_n$ and called a *string* or a *word* over A. The empty sequence is called in this context the *empty string*. A string consisting of a single symbol s is termed an *atomic string*. It will likewise be denoted by s, since it will be clear from the context whether s means a symbol or an atomic string.

Let $\xi\eta$ denote the concatenation of the strings ξ and η. If $\xi = \xi_1\eta\xi_2$ for some strings ξ_1, ξ_2 and $\eta \neq \emptyset$ then η is called a *segment* (or *substring*) of ξ, termed a *proper* segment in case $\eta \neq \xi$. If $\xi_1 = \emptyset$ then η is called an *initial*, if $\xi_2 = \emptyset$, a *terminal* segment of ξ.

Subsets $P, Q, R, \ldots \subseteq M^n$ are called *n-ary predicates of M* or n-ary *relations*. A unary predicate will be identified with the corresponding subset of M. We may write $P\vec{a}$ for $\vec{a} \in P$, and $\neg P\vec{a}$ for $\vec{a} \notin P$. Metatheoretical predicates (or properties) cast in words will often be distinguished from the surrounding text by single quotes, for instance, if we speak of the syntactic predicate 'The variable x occurs in the formula α'. We can do so since quotes inside quotes will not occur in this book. Single-quoted properties are often used in induction principles or reflected in a theory, while ordinary ("double") quotes have a stylistic function only.

An n-ary *operation of M* is a function $f\colon M^n \to M$. Since $M^0 = \{\emptyset\}$, a 0-ary operation of M is of the form $\{(\emptyset, c)\}$, with $c \in M$; it is denoted by

c for short and called a *constant*. Each operation $f\colon M^n \to M$ is uniquely described by the *graph of* f, defined as

$$\operatorname{graph} f := \{(a_1,\dots,a_{n+1}) \in M^{n+1} \mid f(a_1,\dots,a_n) = a_{n+1}\}.$$[1]

Both f and $\operatorname{graph} f$ are essentially the same, but in most situations it is more convenient to distinguish between them.

The most important operations are binary ones. The corresponding symbols are mostly written between the arguments, as in the following listing of properties of a binary operation $\circ$ on a set A. $\circ\colon A^2 \to A$ is

commutative	if	$a \circ b = b \circ a$ for all $a, b \in A$,
associative	if	$a \circ (b \circ c) = (a \circ b) \circ c$ for all $a, b, c \in A$,
idempotent	if	$a \circ a = a$ for all $a \in A$,
invertible	if	for all $a, b \in A$ there are $x, y \in A$ with $a \circ x = b$ and $y \circ a = b$.

If H, Θ (read *eta, theta*) are expressions of our metalanguage, $\mathrm{H} \Leftrightarrow \Theta$ stands for 'H *iff* Θ' which abbreviates 'H *if and only if* Θ'. Similarly, $\mathrm{H} \Rightarrow \Theta$ and H & Θ mean '*if* H *then* Θ' and 'H *and* Θ', respectively, and $\mathrm{H} \vee \Theta$ is to mean 'H *or* Θ.' This notation does not aim at formalizing the metalanguage but serves improved organization of metatheoretic statements. We agree that $\Rightarrow, \Leftrightarrow, \dots$ separate stronger than linguistic binding particles such as "there is" or "for all." Therefore, in the statement

'$X \vdash \alpha \Leftrightarrow X \vDash \alpha$, *for all* X *and all* α' (Theorem 1.4.6)

the comma should not be dropped; otherwise, some serious misunderstanding may arise: '$X \vDash \alpha$ *for all* X *and all* α' is simply false.

$\mathrm{H} :\Leftrightarrow \Theta$ means that the expression H is defined by Θ. When integrating formulas in the colloquial metalanguage, one may use certain abbreviating notation. For instance, '$\alpha \equiv \beta$ and $\beta \equiv \gamma$' is occasionally shortened to $\alpha \equiv \beta \equiv \gamma$. ('the formulas α, β, and β, γ are equivalent'). This is allowed, since in this book the symbol $\equiv$ will never belong to the formal language from which the formulas α, β, γ are taken. W.l.o.g. or w.l.o.g. is a colloquial shorthand of "without loss of generality" used in mathematics.

[1] This means that the left-hand term $\operatorname{graph} f$ is *defined* by the right-hand term. A corresponding meaning has $:=$ throughout, except in programs and flow diagrams, where $x := t$ means the allocation of the value of the term t to the variable x.

Chapter 1

Propositional Logic

Propositional logic, by which we here mean two-valued propositional logic, arises from analyzing connections of given sentences A, B, such as

$$A \textit{ and } B, \quad A \textit{ or } B, \quad \textit{not } A, \quad \textit{if } A \textit{ then } B.$$

These connection operations can be approximately described by two-valued logic. There are other connections that have temporal or local features, for instance, *first A then B* or *here A there B*, as well as unary modal operators like *it is necessarily true that*, whose analysis goes beyond the scope of two-valued logic. These operators are the subject of temporal, modal, or other subdisciplines of many-valued or nonclassical logic. Furthermore, the connections that we began with may have a meaning in other versions of logic that two-valued logic only incompletely captures. This pertains in particular to their meaning in natural or everyday language, where meaning may strongly depend on context.

In two-valued propositional logic such phenomena are set aside. This approach not only considerably simplifies matters, but has the advantage of presenting many concepts, for instance those of consequence, rule induction, or resolution, on a simpler and more perspicuous level. This will in turn save a lot of writing in Chapter **2** when we consider the corresponding concepts in the framework of predicate logic.

We will not consider everything that would make sense in two-valued propositional logic, such as two-valued fragments and problems of definability and interpolation. The reader is referred instead to [KK] or [Ra1]. We will concentrate our attention more on propositional calculi. While there exists a multitude of applications of propositional logic, we will not consider technical applications such as the designing of Boolean circuits and problems of optimization. These topics have meanwhile been integrated into computer science. Rather, some useful applications of the propositional compactness theorem are described comprehensively.

1.1 Boolean Functions and Formulas

Two-valued logic is based on two foundational principles: the *principle of bivalence*, which allows only two truth values, namely *true* and *false*, and the *principle of extensionality*, according to which the truth value of a connected sentence depends only on the truth values of its parts, not on their meaning. Clearly, these principles form only an idealization of the actual relationships.

Questions regarding degrees of truth or the sense-content of sentences are ignored in two-valued logic. Despite this simplification, or indeed because of it, such a method is scientifically successful. One does not even have to know exactly what the truth values *true* and *false* actually are. Indeed, in what follows we will identify them with the two symbols 1 and 0. Of course, one could have chosen any other apt symbols such as $\top$ and $\bot$ or `t` and `f`. The advantage here is that all conceivable interpretations of *true* and *false* remain open, including those of a purely technical nature, for instance the two states of a gate in a Boolean circuit.

According to the meaning of the word *and*, the conjunction *A and B* of sentences A, B, in formalized languages written as $A \wedge B$ or $A \,\&\, B$, is true if and only if A, B are both true and is false otherwise. So conjunction corresponds to a binary function or operation over the set $\{0,1\}$ of truth values, named the $\wedge$*-function* and denoted by $\wedge$. It is given by its *value matrix* $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$, where, in general, $\begin{pmatrix} 1\circ 1 & 1\circ 0 \\ 0\circ 1 & 0\circ 0 \end{pmatrix}$ represents the value matrix or *truth table* of a binary function $\circ$ with arguments and values in $\{0,1\}$. The delimiters of these small matrices will usually be omitted.

A function $f\colon \{0,1\}^n \to \{0,1\}$ is called an n-ary *Boolean function* or *truth function*. Since there are 2^n n-tuples of $0,1$, it is easy to see that the number of n-ary Boolean functions is 2^{2^n}. We denote their totality by $\boldsymbol{B}_n$. While $\boldsymbol{B}_2$ has $2^4 = 16$ members, there are only four unary Boolean functions. One of these is *negation*, denoted by $\neg$ and defined by $\neg 1 = 0$ and $\neg 0 = 1$. $\boldsymbol{B}_0$ consists just of the constants 0 and 1.

The first column of the table below contains the common binary connections with examples of their instantiation in English. The second column lists some of its traditional symbols, which also denote the corresponding truth function, and the third its truth table. *Disjunction* is the *inclusive or* and is to be distinguished from the *exclusive disjunction*. The latter corresponds to addition modulo 2 and is therefore given the symbol $+$. In Boolean circuits the functions $+, \downarrow, \uparrow$ are often denoted by *xor*, *nor*, and *nand*; the latter is also known as the *Sheffer function*. Recall our agreement in the section *Notation* that the symbols &, $\vee$, $\Rightarrow$, and $\Leftrightarrow$ will be used only on the metatheoretic level.

A connected sentence and its corresponding truth function need not be denoted by the same symbol; for example, one might take $\land$ for conjunction and *et* as the corresponding truth function. But in doing so one would only be creating extra notation, but no new insights. The meaning of a symbol will always be clear from the context: if α, β are sentences of a formal language, then $\alpha \land \beta$ denotes their conjunction; if a, b are truth values, then $a \land b$ just denotes a truth value. Occasionally, we may want to refer to the symbols $\land, \lor, \neg, \dots$ themselves, setting their meaning temporarily aside. Then we talk of the *connectives* or *truth functors* $\land, \lor, \neg, \dots$

compound sentence	**symbol**	**truth table**
conjunction *A and B*; *A as well as B*	$\land$, &	1 0 0 0
disjunction *A or B*	$\lor$, $\vee$	1 1 1 0
implication *if A then B; B provided A*	$\to$, $\Rightarrow$	1 0 1 1
equivalence *A if and only if B; A iff B*	$\leftrightarrow$, $\Leftrightarrow$	1 0 0 1
exclusive disjunction *either A or B but not both*	$+$	0 1 1 0
nihilation *neither A nor B*	$\downarrow$	0 0 0 1
incompatibility *not at once A and B*	$\uparrow$	0 1 1 1

Sentences formed using connectives given in the table are said to be logically equivalent if their corresponding truth tables coincide. This is the case, e.g., for the sentences A *provided* B and A *or not* B, which represent the *converse implication*, denoted by $A \leftarrow B$.[1] It does not appear in the table, since it arises by swapping A, B in the implication. This and similar reasons explain why only a few of the sixteen binary Boolean functions require notation. Some other examples of logical equivalent sentences are *if* A *and* B *then* C, C *provided* A *and* B, and A *and* B *only if* C.

In order to recognize and describe logical equivalence of compound sentences it is useful to create a suitable formalism or a formal language. The idea is basically the same as in arithmetic, where general statements are more clearly expressed by means of certain formulas. As with arithmetical terms, we consider propositional formulas as strings of signs built in given ways from basic symbols. Among these basic symbols are variables, for our purposes called *propositional variables*, the set of which is denoted by *PV*. Traditionally, these are symbolized by $p_0, p_1, \ldots$ However, our numbering of the variables below begins with p_1 rather than with p_0, enabling us later on to represent Boolean functions more conveniently. Further, we use certain logical signs such as $\wedge, \vee, \neg, \ldots$, similar to the signs $+, \cdot, \ldots$ of arithmetic. Finally, parentheses (,) will serve as technical aids, although these are dispensable, as will be seen later on.

Each time a propositional language is in question, the set of its logical symbols, called the *logical signature*, and the set of its variables must be given in advance. For instance, it is crucial in some applications of propositional logic in Section **1.5** for *PV* to be an arbitrary set, and not a countably infinite one as indicated previously. Put concretely, we define a propositional language $\mathcal{F}$ of formulas built up from the symbols $(\,,\,)\,, \wedge\,, \vee\,, \neg\,, p_1, p_2, \ldots$ inductively as follows:

(F1) The atomic strings $p_1, p_2, \ldots$ are formulas, called *prime formulas*, also called *atomic* formulas, or simply *prime.*

(F2) If the strings α, β are formulas, then so too are the strings $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, and $\neg\alpha$.

This is a recursive (somewhat sloppily also called inductive) definition in the set of strings on the alphabet of the mentioned symbols, that is,

[1] Converse implication is used in the programming language PROLOG, see **4.6**.

only those strings gained using (F1) or (F2) are in this context formulas. Stated set-theoretically, $\mathcal{F}$ is the smallest (i.e., the intersection) of all sets of strings S built from the aforementioned symbols with the properties

$$\text{(f1) } p_1, p_2, \ldots \in S, \quad \text{(f2) } \alpha, \beta \in S \;\Rightarrow\; (\alpha \wedge \beta), (\alpha \vee \beta), \neg\alpha \in S.$$

Example. $(p_1 \wedge (p_2 \vee \neg p_1))$ is a formula. On the other hand, its initial segment $(p_1 \wedge (p_2 \vee \neg p_1)$ is not, because a closing parenthesis is missing. It is intuitively clear and will rigorously be proved on the next page that the number of left parentheses occurring in a formula coincides with the number of its right parentheses.

Remark 1. (f1) and (f2) are set-theoretic translations of (F1) and (F2). Some authors like to add a third condition to (F1), (F2), namely (F3): *No other strings than those obtained by* (F1) *and* (F2) *are formulas in this context.* But this at most underlines that (F1), (F2) are the only formula-building rules; (F3) *follows* from our definition, as its set-theoretic translation by (f1), (f2) indicates. Note that we do not strictly distinguish between the symbol p_i and the prime formula or atomic string p_i. Note also that in the formula definition parentheses are needed only for binary connectives, not if a formula starts with $\neg$. By a slightly more involved definition at least the outermost parentheses in formulas of the form $(\alpha \circ \beta)$ with a binary connective $\circ$ could be saved. Howsoever propositional formulas are defined, what counts is their unique readability, see page 7.

The formulas defined by (F1), (F2) are called *Boolean formulas*, because they are obtained using the *Boolean signature* $\{\wedge, \vee, \neg\}$. Should further connectives belong to the logical signature, for example $\rightarrow$ or $\leftrightarrow$, (F2) of the above definition must be augmented accordingly. But unless stated otherwise, $(\alpha \rightarrow \beta)$ and $(\alpha \leftrightarrow \beta)$ are here just abbreviations; the first is $\neg(\alpha \wedge \neg\beta)$, the second is $((\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha))$.

Occasionally, it is useful to have symbols in the logical signature for always false and always true, $\bot$ and $\top$ respectively, say, called *falsum* and *verum* and sometimes also denoted by 0 and 1. These are to be regarded as supplementary prime formulas, and clause (F1) should be altered accordingly. However, we prefer to treat $\bot$ and $\top$ as abbreviations: $\bot := (p_1 \wedge \neg p_1)$ and $\top := \neg\bot$.

For the time being we let $\mathcal{F}$ be the set of all Boolean formulas, although everything said about $\mathcal{F}$ holds correspondingly for any propositional language. Propositional variables will henceforth be denoted by $p, q, \ldots$, formulas by $\alpha, \beta, \gamma, \delta, \varphi, \ldots$, prime formulas also by π, and sets of propositional formulas by X, Y, Z, where these letters may also be indexed.

For the reason of parenthesis economy in formulas, we set some conventions similar to those used in writing arithmetical terms.

1. The outermost parentheses in a formula may be omitted (if there are any). For example, $(p \vee q) \wedge \neg p$ may be written in place of $((p \vee q) \wedge \neg p)$. Note that $(p \vee q) \wedge \neg p$ is not itself a formula but *denotes* the formula $((p \vee q) \wedge \neg p)$.

2. In the order $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, each connective binds more strongly than those following it. Thus, one may write $p \vee q \wedge \neg p$ instead of $p \vee (q \wedge \neg p)$, which means $(p \vee (q \wedge \neg p))$ by convention 1.

3. By the multiple use of $\rightarrow$ we *associate to the right*. So $p \rightarrow q \rightarrow p$ is to mean $p \rightarrow (q \rightarrow p)$. Multiple occurrences of other binary connectives are associated to the left, for instance, $p \wedge q \wedge \neg p$ means $(p \wedge q) \wedge \neg p$. In place of $\alpha_0 \wedge \cdots \wedge \alpha_n$ and $\alpha_0 \vee \cdots \vee \alpha_n$ we may write $\bigwedge_{i \leqslant n} \alpha_i$ and $\bigvee_{i \leqslant n} \alpha_i$, respectively.

Also, in arithmetic, one normally associates to the left. An exception is the term x^{y^z}, where traditionally association to the right is used, that is, x^{y^z} equals $x^{(y^z)}$. Association to the right has some advantages in writing tautologies in which $\rightarrow$ occurs several times; for instance in the examples of tautologies listed in **1.3** on page 18.

The above conventions are based on a reliable syntax in the framework of which intuitively clear facts, such as the identical number of left and right parentheses in a formula, are rigorously provable. These proofs are generally carried out using induction on the construction of a formula. To make this clear we denote by $\mathcal{E}\varphi$ that a property $\mathcal{E}$ holds for a string φ. For example, let $\mathcal{E}$ mean the property 'φ is a formula that has equally many right- and left-hand parentheses'. $\mathcal{E}$ is trivially valid for prime formulas, and if $\mathcal{E}\alpha$, $\mathcal{E}\beta$ then clearly also $\mathcal{E}(\alpha \wedge \beta)$, $\mathcal{E}(\alpha \vee \beta)$, and $\mathcal{E}\neg\alpha$. From this we may conclude that $\mathcal{E}$ applies to all formulas, our reasoning being a particularly simple instance of the following

Principle of formula induction. *Let $\mathcal{E}$ be a property of strings that satisfies the conditions*

(o) *$\mathcal{E}\pi$ for all prime formulas π,*

(s) *$\mathcal{E}\alpha, \mathcal{E}\beta \Rightarrow \mathcal{E}(\alpha \wedge \beta), \mathcal{E}(\alpha \vee \beta), \mathcal{E}\neg\alpha$, for all $\alpha, \beta \in \mathcal{F}$.*

Then $\mathcal{E}\varphi$ holds for all formulas φ.

The justification of this principle is straightforward. The set S of all strings with property $\mathcal{E}$ has, thanks to (o) and (s), the properties (f1) and (f2) on page 5. But $\mathcal{F}$ is the smallest such set. Therefore, $\mathcal{F} \subseteq S$. In words, $\mathcal{E}$ applies to all formulas φ. Clearly, if other connectives are involved, condition (s) must accordingly be modified.

It is intuitively clear and easily confirmed inductively on φ that a compound Boolean formula φ (i.e., φ is not prime) is of the form $\varphi = \neg\alpha$ or $\varphi = (\alpha \wedge \beta)$ or $\varphi = (\alpha \vee \beta)$ for suitable $\alpha, \beta \in \mathcal{F}$. Moreover, this decomposition is unique. For instance, $(\alpha \wedge \beta)$ cannot at the same time be written $(\alpha' \vee \beta')$ with perhaps different formulas α', β'. Thus, compound formulas have the unique readability property, more precisely, the

Unique formula reconstruction property. *Each compound formula $\varphi \in \mathcal{F}$ is either of the form $\neg\alpha$ or $(\alpha \circ \beta)$ for some uniquely determined formulas $\alpha, \beta \in \mathcal{F}$, where $\circ$ is either $\wedge$ or $\vee$.*

This property is less obvious than it might seem. Nonetheless, the proof is left as an exercise (Exercise 4) in order to maintain the flow of things. It may be a surprise to the novice that for the unique formula reconstruction, parentheses are dispensable throughout. Indeed, propositional formulas, like arithmetical terms, can be written without any parentheses; this is realized in *Polish notation* (= PN), also called *prefix notation*, once widely used in the logic literature. The idea consists in altering (F2) as follows: *if α, β are formulas then so too are $\wedge\alpha\beta$, $\vee\alpha\beta$, and $\neg\alpha$.* Similar to PN is RPN (*reverse Polish notation*), still used in some programming languages like PostScript. RPN differs from PN only in that a connective is placed *after* the arguments. For instance, $(p \wedge (q \vee \neg p))$ is written in RPN as $pqp\neg\vee\wedge$. Reading PN or RPN requires more effort due to the high density of information; but by the same token it can be processed very fast by a computer or a high-tech printer getting its job as a PostScript program. The only advantage of the parenthesized version is that its decoding is somewhat easier for our eye through the dilution of information.

Intuitively it is clear what a *subformula* of a formula φ is; for example, $(q \wedge \neg p)$ is a subformula of $(p \vee (q \wedge \neg p))$. All the same, for some purposes it is convenient to characterize the set $\mathrm{Sf}\,\varphi$ of all subformulas of φ inductively:

$$\mathrm{Sf}\,\pi = \{\pi\} \text{ for prime formulas } \pi; \quad \mathrm{Sf}\,\neg\alpha = \mathrm{Sf}\,\alpha \cup \{\neg\alpha\},$$
$$\mathrm{Sf}(\alpha \circ \beta) = \mathrm{Sf}\,\alpha \cup \mathrm{Sf}\,\beta \cup \{(\alpha \circ \beta)\} \text{ for a binary connective } \circ.$$

Thus, a formula is always regarded as a subformula of itself. The above is a typical example of a *recursive definition on the construction of formulas*. Another example of such a definition is the *rank* $\operatorname{rk}\varphi$ of a formula φ, which provides a sometimes more convenient measure of the complexity of φ than its length as a string and occasionally simplifies inductive arguments. Intuitively, $\operatorname{rk}\varphi$ is the highest number of nested connectives in φ. Let $\operatorname{rk}\pi = 0$ for prime formulas π, and if $\operatorname{rk}\alpha$ and $\operatorname{rk}\beta$ are already defined, then $\operatorname{rk}\neg\alpha = \operatorname{rk}\alpha + 1$ and $\operatorname{rk}(\alpha\circ\beta) = \max\{\operatorname{rk}\alpha, \operatorname{rk}\beta\} + 1$. Here $\circ$ denotes any binary connective. We will not give here a general formulation of this definition procedure because it is very intuitive and similar to the well-known procedure of recursive definitions on $\mathbb{N}$. It has been made sufficiently clear by the preceding examples. Its justification is based on the unique reconstruction property and insofar not quite trivial, in contrast to the proof procedure by induction on formulas that immediately follows from the definition of propositional formulas.

If a property is to be proved by induction on the construction of formulas φ, we will say that it is a *proof by induction on* φ. Similarly, the recursive construction of a function f on $\mathcal{F}$ will generally be referred to as *defining* f *by recursion on* φ, often somewhat sloppily paraphrased as *defining* f *by induction on* φ. Examples are Sf and rk. Others will follow.

Since the truth value of a connected sentence depends only on the truth values of its constituent parts, we may assign to every propositional variable of α a truth value rather than a sentence, thereby evaluating α, i.e., calculating a truth value. Similarly, terms are evaluated in, say, the arithmetic of real numbers, whose value is then a real (= real number). An arithmetical term t in the variables $x_1, \dots, x_n$ describes an n-ary function whose arguments and values are reals, while a formula φ in $p_1, \dots, p_n$ describes an n-ary Boolean function. To be precise, a propositional *valuation*, or alternatively, a (propositional) *model*, is a mapping $w\colon PV \to \{0,1\}$ that can also be understood as a mapping from the set of prime formulas to $\{0,1\}$. We can extend this to a mapping from the whole of $\mathcal{F}$ to $\{0,1\}$ (likewise denoted by w) according to the stipulations

$$(*)\quad w(\alpha\wedge\beta) = w\alpha\wedge w\beta;\quad w(\alpha\vee\beta) = w\alpha\vee w\beta;\quad w\neg\alpha = \neg w\alpha.$$ [2]

By *the value* $w\varphi$ *of a formula* φ under a valuation $w\colon PV \to \{0,1\}$

[2] We often use $(*)$ or $(\star)$ as a temporary label for a condition (or property) that we refer back to in the text following the labeled condition.

we mean the value given by this extension. We could denote the extended mapping by $\hat{w}$, say, but it is in fact not necessary to distinguish it symbolically from $w\colon PV \to \{0,1\}$ because the latter determines the extension uniquely. Similarly, we keep the same symbol if an operation in $\mathbb{N}$ extends to a larger domain. If the logical signature contains further connectives, for example $\to$, then $(*)$ must be supplemented accordingly, with $w(\alpha \to \beta) = w\alpha \to w\beta$ in the example. However, if $\to$ is defined as in the Boolean case, then this equation must be provable. Indeed, it is provable, because from our definition of $\alpha \to \beta$ it follows that

$$w(\alpha \to \beta) = w\neg(\alpha \wedge \neg\beta) = \neg w(\alpha \wedge \neg\beta) = \neg(w\alpha \wedge \neg w\beta) = w\alpha \to w\beta,$$

for any w. A corresponding remark could be made with respect to $\leftrightarrow$ and to $\top$ and $\bot$. Always $w\top = 1$ and $w\bot = 0$ by our definition of $\top, \bot$, in accordance with the meaning of these symbols. However, if these or similar symbols belong to the logical signature, then suitable equations must be added to the definition of w.

Let $\mathcal{F}_n$ denote the set of all formulas of $\mathcal{F}$ in which at most the variables $p_1, \ldots, p_n$ occur $(n > 0)$. Then it can easily be seen that $w\alpha$ for the formula $\alpha \in \mathcal{F}_n$ depends only on the truth values of $p_1, \ldots, p_n$. In other words, $\alpha \in \mathcal{F}_n$ satisfies for all valuations w, w',

$$(\star) \quad w\alpha = w'\alpha \text{ whenever } wp_i = w'p_i \text{ for } i = 1, \ldots, n.$$

The simple proof of $(\star)$ follows from induction on the construction of formulas in $\mathcal{F}_n$, observing that these are closed under the operations $\neg, \wedge, \vee$. Clearly, $(\star)$ holds for $p \in \mathcal{F}_n$, and if $(\star)$ is valid for $\alpha, \beta \in \mathcal{F}_n$, then also for $\neg\alpha$, $\alpha \wedge \beta$, and $\alpha \vee \beta$. It is then clear that each $\alpha \in \mathcal{F}_n$ defines or represents an n-ary Boolean function according to the following

Definition. $\alpha \in \mathcal{F}_n$ *represents* the function $f \in \boldsymbol{B}_n$ (or f is represented by α) whenever $w\alpha = fw\vec{p}$ $(:= f(wp_1, \ldots, wp_n))$ for all valuations w.

Because $w\alpha$ for $\alpha \in \mathcal{F}_n$ is uniquely determined by $wp_1, \ldots, wp_n$, α represents precisely one function $f \in \boldsymbol{B}_n$, sometimes written as $\alpha^{(n)}$. For instance, both $p_1 \wedge p_2$ and $\neg(\neg p_1 \vee \neg p_2)$ represent the $\wedge$-function, as can easily be illustrated using a table. Similarly, $\neg p_1 \vee p_2$ and $\neg(p_1 \wedge \neg p_2)$ represent the $\to$-function, and $p_1 \vee p_2$, $\neg(\neg p_1 \wedge \neg p_2)$, $(p_1 \to p_2) \to p_2$ all represent the $\vee$-function. Incidentally, the last formula shows that the $\vee$-connective can be expressed using implication alone.

There is a caveat though: since $\alpha = p_1 \vee p_2$, for instance, belongs not only to $\mathcal{F}_2$ but to $\mathcal{F}_3$ as well, α also represents the Boolean function $f\colon (x_1, x_2, x_3) \mapsto x_1 \vee x_2$. However, the third argument is only "fictional," or put another way, the function f is not essentially ternary.

In general we say that an operation $f\colon M^n \to M$ is *essentially n-ary* if f has no fictional arguments, where the ith argument of f is called *fictional* whenever for all $x_1, \ldots, x_i, \ldots x_n \in M$ and all $x_i' \in M$,

$$f(x_1, \ldots, x_i, \ldots, x_n) = f(x_1, \ldots, x_i', \ldots, x_n).$$

Identity and the $\neg$-function are the essentially unary Boolean functions, and out of the sixteen binary functions, only ten are essentially binary, as is seen in scrutinizing the possible truth tables.

Remark 2. If a_n denotes temporarily the number of all n-ary Boolean functions and e_n the number of all essentially n-ary Boolean functions, it is not particularly difficult to prove that $a_n = \sum_{i \leqslant n} \binom{n}{i} e_i$. Solving for e_n results in $e_n = \sum_{i \leqslant n} (-1)^{n-i} \binom{n}{i} a_i$. However, we will not make use of these equations. These become important only in a more specialized study of Boolean functions.

Exercises

1. $f \in \boldsymbol{B}_n$ is called *linear* if $f(x_1, \ldots, x_n) = a_0 + a_1 x_1 + \cdots + a_n x_n$ for suitable coefficients $a_0, \ldots, a_n \in \{0, 1\}$. Here $+$ denotes exclusive disjunction (addition modulo 2) and the not written multiplication is conjunction (i.e., $a_i x_i = x_i$ for $a_i = 1$ and $a_i x_i = 0$ for $a_i = 0$). (a) Show that the above representation of a linear function f is unique. (b) Determine the number of n-ary linear Boolean functions. (c) Prove that each formula α in $\neg, +$ (i.e., α is a formula of the logical signature $\{\neg, +\}$) represents a linear Boolean function.

2. Verify that a compound Boolean formula φ is either of the form $\varphi = \neg\alpha$ or else $\varphi = (\alpha \wedge \beta)$ or $\varphi = (\alpha \vee \beta)$ for suitable formulas α, β (this is the easy part of the unique reconstruction property).

3. Prove that a proper initial segment of a formula φ is never a formula. Equivalently: If $\alpha\xi = \beta\eta$ with $\alpha, \beta \in \mathcal{F}$ and arbitrary strings ξ, η, then $\alpha = \beta$. The same holds for formulas in PN, but not in RPN.

4. Prove (with Exercise 3) the second more difficult part of the unique reconstruction property, the claim of uniqueness.

1.2 Semantic Equivalence and Normal Forms

Throughout this chapter w will always denote a propositional valuation. Formulas α, β are called (logically or semantically) *equivalent*, and we write $\alpha \equiv \beta$, when $w\alpha = w\beta$ for all valuations w. For example $\alpha \equiv \neg\neg\alpha$. Obviously, $\alpha \equiv \beta$ iff for any n such that $\alpha, \beta \in \mathcal{F}_n$, both formulas represent the same n-ary Boolean function. It follows that at most 2^{2^n} formulas in $\mathcal{F}_n$ can be pairwise inequivalent, since there are no more than 2^{2^n} n-ary Boolean functions.

In arithmetic one writes simply $s = t$ to express that the terms s, t represent the same function. For example, $(x+y)^2 = x^2 + 2xy + y^2$ expresses the equality of values of the left- and right-hand terms for all $x, y \in \mathbb{R}$. This way of writing is permissible because formal syntax plays a minor role in arithmetic. In formal logic, however, as is always the case when syntactic considerations are to the fore, one uses the equality sign in messages like $\alpha = \beta$ only for the syntactic identity of the strings α and β. Therefore, the equivalence of formulas must be denoted differently. Clearly, for all formulas α, β, γ the following equivalences hold:

$$\begin{array}{rcllcll}
\alpha\wedge(\beta\wedge\gamma) & \equiv & \alpha\wedge\beta\wedge\gamma, & \alpha\vee(\beta\vee\gamma) & \equiv & \alpha\vee\beta\vee\gamma & \text{(associativity);} \\
\alpha\wedge\beta & \equiv & \beta\wedge\alpha, & \alpha\vee\beta & \equiv & \beta\vee\alpha & \text{(commutativity);} \\
\alpha\wedge\alpha & \equiv & \alpha, & \alpha\vee\alpha & \equiv & \alpha & \text{(idempotency);} \\
\alpha\wedge(\alpha\vee\beta) & \equiv & \alpha, & \alpha\vee\alpha\wedge\beta & \equiv & \alpha & \text{(absorption);} \\
\alpha\wedge(\beta\vee\gamma) & \equiv & \alpha\wedge\beta\vee\alpha\wedge\gamma, & & & & (\wedge\text{-distributivity);} \\
\alpha\vee\beta\wedge\gamma & \equiv & (\alpha\vee\beta)\wedge(\alpha\vee\gamma) & & & & (\vee\text{-distributivity);} \\
\neg(\alpha\wedge\beta) & \equiv & \neg\alpha\vee\neg\beta, & \neg(\alpha\vee\beta) & \equiv & \neg\alpha\wedge\neg\beta & \text{(de Morgan rules).}
\end{array}$$

Furthermore, $\alpha\vee\neg\alpha \equiv \top$, $\alpha\wedge\neg\alpha \equiv \bot$, and $\alpha\wedge\top \equiv \alpha\vee\bot \equiv \alpha$. It is also useful to list certain equivalences for formulas containing $\to$, for example the frequently used $\alpha\to\beta \equiv \neg\alpha\vee\beta$ ($\equiv \neg(\alpha\wedge\neg\beta)$), and the important

$$\alpha\to\beta\to\gamma \equiv \alpha\wedge\beta\to\gamma \equiv \beta\to\alpha\to\gamma.$$

To generalize: $\alpha_1 \to \cdots \to \alpha_n \equiv \alpha_1\wedge\cdots\wedge\alpha_{n-1}\to\alpha_n$. Further, we mention the "left distributivity" of implication with respect to $\wedge$ and $\vee$, namely

$$\alpha\to\beta\wedge\gamma \equiv (\alpha\to\beta)\wedge(\alpha\to\gamma); \quad \alpha\to\beta\vee\gamma \equiv (\alpha\to\beta)\vee(\alpha\to\gamma).$$

Should the symbol $\to$ lie to the right then the following are valid:

$$\alpha\wedge\beta\to\gamma \equiv (\alpha\to\gamma)\vee(\beta\to\gamma); \quad \alpha\vee\beta\to\gamma \equiv (\alpha\to\gamma)\wedge(\beta\to\gamma).$$

Remark 1. These last two logical equivalences are responsible for a curious phenomenon in everyday language. For example, the two sentences

A: *Students and pensioners pay half price*,
B: *Students or pensioners pay half price*

evidently have the same meaning. How to explain this? Let *student* and *pensioner* be abbreviated by S, P, and *pay half price* by H. Then

$$\alpha: \quad (S \to H) \wedge (P \to H), \qquad \beta: \quad (S \vee P) \to H$$

express somewhat more precisely the factual content of A and B, respectively. Now, according to our truth tables, the formulas α and β are simply logically equivalent. The everyday-language statements A and B of α and β obscure the structural difference of α and β through an apparently synonymous use of the words *and* and *or*.

Obviously, $\equiv$ is an equivalence relation, that is,

$$\begin{array}{rl} \alpha \equiv \alpha & \text{(reflexivity)}, \\ \alpha \equiv \beta \;\Rightarrow\; \beta \equiv \alpha & \text{(symmetry)}, \\ \alpha \equiv \beta, \beta \equiv \gamma \;\Rightarrow\; \alpha \equiv \gamma & \text{(transitivity)}. \end{array}$$

Moreover, $\equiv$ is a *congruence relation* on $\mathcal{F}$,[3] i.e., for all $\alpha, \alpha', \beta, \beta'$,

$$\alpha \equiv \alpha', \beta \equiv \beta' \;\Rightarrow\; \alpha \circ \beta \equiv \alpha' \circ \beta', \neg\alpha \equiv \neg\alpha' \qquad (\circ \in \{\wedge, \vee\}).$$

For this reason the *replacement theorem* holds: $\alpha \equiv \alpha' \;\Rightarrow\; \varphi \equiv \varphi'$, where φ' is obtained from φ by replacing one or several of the possible occurrences of the subformula α in φ by α'. For instance, by replacing the subformula $\neg p \vee \neg q$ by the equivalent formula $\neg(p \wedge q)$ in $\varphi = (\neg p \vee \neg q) \wedge (p \vee q)$ we obtain $\varphi' = \neg(p \wedge q) \wedge (p \vee q)$, which is equivalent to φ. A similar replacement theorem also holds for arithmetical terms and is constantly used in their manipulation. This mostly goes unnoticed, because $=$ is written instead of $\equiv$, and the replacement for $=$ is usually correctly applied. The simple inductive proof of the replacement theorem will be given in a somewhat broader context in **2.4**.

Furnished with the equivalences $\neg\neg\alpha \equiv \alpha$, $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$, and $\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$, and using replacement it is easy to construct for each formula an equivalent formula in which $\neg$ stands only in front of variables. For example, $\neg(p \wedge q \vee r) \equiv \neg(p \wedge q) \wedge \neg r \equiv (\neg p \vee \neg q) \wedge \neg r$ is obtained in this way. This observation follows also from Theorem 2.1.

[3] This concept, stemming originally from geometry, is meaningfully defined in every algebraic structure and is one of the most important and most general mathematical concepts; see **2.1**. The definition is equivalent to the condition

$$\alpha \equiv \alpha' \;\Rightarrow\; \alpha \circ \beta \equiv \alpha' \circ \beta, \beta \circ \alpha \equiv \beta \circ \alpha', \neg\alpha \equiv \neg\alpha', \text{ for all } \alpha, \alpha', \beta.$$

It is always something of a surprise to the newcomer that independent of its arity, *every* Boolean function can be represented by a Boolean formula. While this can be proved in various ways, we take the opportunity to introduce certain normal forms and therefore begin with the following

Definition. Prime formulas and negations of prime formulas are called *literals*. A disjunction $\alpha_1\vee\cdots\vee\alpha_n$, where each α_i is a conjunction of literals, is called a *disjunctive normal form*, a DNF for short. A conjunction $\beta_1\wedge\cdots\wedge\beta_n$, where every β_i is a disjunction of literals, is called a *conjunctive normal form*, a CNF for short.

Example 1. The formula $p\vee(q\wedge\neg p)$ is a DNF; $p\vee q$ is at once a DNF and a CNF; $p\vee\neg(q\wedge\neg p)$ is neither a DNF nor a CNF.

Theorem 2.1 states that every Boolean function is represented by a Boolean formula, indeed by a DNF, and also by a CNF. It would suffice to show that for given n there are at least 2^{2^n} pairwise inequivalent DNFs (resp. CNFs). However, we present instead a constructive proof whereby for a Boolean function given in tabular form a representing DNF (resp. CNF) can explicitly be written down. In Theorem 2.1 we temporarily use the following notation: $p^1 := p$ and $p^0 := \neg p$. With this stipulation, $w(p_1^{x_1}\wedge p_2^{x_2}) = 1$ iff $wp_1 = x_1$ and $wp_2 = x_2$. More generally, induction on $n \geqslant 1$ easily shows that for all $x_1,\dots,x_n\in\{0,1\}$,

$$(*)\quad w(p_1^{x_1}\wedge\cdots\wedge p_n^{x_n}) = 1 \Leftrightarrow w\vec{p} = \vec{x}\quad \text{(i.e., } wp_1 = x_1,\dots,wp_n = x_n\text{)}.$$

Theorem 2.1. *Every Boolean function f with $f\in\boldsymbol{B}_n$ $(n>0)$ is representable by a* DNF, *namely by*

$$\alpha_f := \bigvee_{f\vec{x}=1} p_1^{x_1}\wedge\cdots\wedge p_n^{x_n}.^{4}$$

At the same time, f is representable by the CNF

$$\beta_f := \bigwedge_{f\vec{x}=0} p_1^{\neg x_1}\vee\cdots\vee p_n^{\neg x_n}.$$

Proof. By the definition of α_f, the following equivalences hold for an arbitrary valuation w:

[4] The disjuncts of α_f can be arranged, for instance, according to the lexicographical order of the n-tuples $(x_1,\dots,x_n)\in\{0,1\}^n$. If the disjunction is empty (that is, if f does not take the value 1) let α_f be $\bot$ $(= p_1\wedge\neg p_1)$. Thus, the empty disjunction is $\bot$. Similarly, the empty conjunction equals $\top$ $(=\neg\bot)$. These conventions correspond to those in arithmetic, where the empty sum is 0 and the empty product is 1.

$$\begin{aligned} w\alpha_f = 1 \quad &\Leftrightarrow \quad \text{there is an } \vec{x} \text{ with } f\vec{x} = 1 \text{ and } w(p_1^{x_1} \land \cdots \land p_n^{x_n}) = 1 \\ &\Leftrightarrow \quad \text{there is an } \vec{x} \text{ with } f\vec{x} = 1 \text{ and } w\vec{p} = \vec{x} \quad \big(\text{by } (*)\big) \\ &\Leftrightarrow \quad fw\vec{p} = 1 \quad (\text{replace } \vec{x} \text{ by } w\vec{p}). \end{aligned}$$

Thus, $w\alpha_f = 1 \Leftrightarrow fw\vec{p} = 1$. From this equivalence, and because there are only two truth values, $w\alpha_f = fw\vec{p}$ follows immediately. The representability proof of f by β_f runs analogously; alternatively, Theorem 2.4 below may be used. ❏

Example 2. For the *exclusive-or function* $+$, the construction of α_f in Theorem 2.1 gives the representing DNF $p_1 \land \neg p_2 \lor \neg p_1 \land p_2$, because $(1,0), (0,1)$ are the only pairs for which $+$ has the value 1. The CNF given by the theorem, on the other hand, is $(p_1 \lor p_2) \land (\neg p_1 \lor \neg p_2)$; the equivalent formula $(p_1 \lor p_2) \land \neg(p_1 \land p_2)$ makes the meaning of the exclusive-or compound particularly intuitive.

$p_1 \land p_2 \lor \neg p_1 \land p_2 \lor \neg p_1 \land \neg p_2$ is the DNF given by Theorem 2.1 for the Boolean function $\rightarrow$. It is longer than the formula $\neg p_1 \lor p_2$, which is also a representing DNF. But the former is distinctive in that each of its disjuncts contains each variable occurring in the formula exactly once. A DNF of n variables with the analogous property is called *canonical*. The notion of canonical CNF is correspondingly explained. For instance, the function $\leftrightarrow$ is represented by the canonical CNF $(\neg p_1 \lor p_2) \land (p_1 \lor \neg p_2)$ according to Theorem 2.1, which always provides canonical normal forms as representing formulas.

Since each formula represents a certain Boolean function, Theorem 2.1 immediately implies the following fact, which has also a (more lengthy) syntactical proof with the replacement theorem mentioned on page 12.

Corollary 2.2. *Each $\varphi \in \mathcal{F}$ is equivalent to a* DNF *and to a* CNF.

Functional completeness. A logical signature is called *functional complete* if every Boolean function is representable by a formula in this signature. Theorem 2.1 shows that $\{\neg, \land, \lor\}$ is functional complete. Because of $p \lor q \equiv \neg(\neg p \land \neg q)$ and $p \land q \equiv \neg(\neg p \lor \neg q)$, one can further leave aside $\lor$, or alternatively $\land$. This observation is the content of

Corollary 2.3. *Both $\{\neg, \land\}$ and $\{\neg, \lor\}$ are functional complete.*

Therefore, to show that a logical signature L is functional complete, it is enough to represent $\neg, \land$ or else $\neg, \lor$ by formulas in L. For example, because $\neg p \equiv p \to 0$ and $p \wedge q \equiv \neg(p \to \neg q)$, the signature $\{\to, 0\}$ is functional complete. On the other hand, $\{\to, \wedge, \vee\}$, and a fortiori $\{\to\}$, are not. Indeed, $w\varphi = 1$ for any formula φ in $\to, \wedge, \vee$ and any valuation w such that $wp = 1$ for all p. This can readily be confirmed by induction on φ. Thus, never $\neg p \equiv \varphi$ for any such formula φ.

It is noteworthy that the signature containing only $\downarrow$ is functional complete: from the truth table for $\downarrow$ we get $\neg p \equiv p \downarrow p$ as well as $p \wedge q \equiv \neg p \downarrow \neg q$. Likewise for $\{\uparrow\}$, because $\neg p \equiv p \uparrow p$ and $p \vee q \equiv \neg p \uparrow \neg q$. That $\{\uparrow\}$ must necessarily be functional complete once we know that $\{\downarrow\}$ is will become obvious in the discussion of the duality theorem below. Even up to term equivalence, there still exist infinitely many signatures. Here signatures are called *term equivalent* if the formulas of these signatures represent the same Boolean functions as in Exercise 2, for instance.

Define inductively on the formulas from $\mathcal{F}$ a mapping $\delta : \mathcal{F} \to \mathcal{F}$ by

$$p^\delta = p, \quad (\neg\alpha)^\delta = \neg\alpha^\delta, \qquad (\alpha \wedge \beta)^\delta = \alpha^\delta \vee \beta^\delta, \quad (\alpha \vee \beta)^\delta = \alpha^\delta \wedge \beta^\delta.$$

α^δ is called the *dual formula* of α and is obtained from α simply by interchanging $\wedge$ and $\vee$. Obviously, for a DNF α, α^δ is a CNF, and vice versa. Define the *dual* of $f \in \boldsymbol{B}_n$ by $f^\delta\vec{x} := \neg f \neg\vec{x}$ with $\neg\vec{x} := (\neg x_1, \dots, \neg x_n)$. Clearly $f^{\delta^2} := (f^\delta)^\delta = f$ since $(f^\delta)^\delta\vec{x} = \neg\neg f \neg\neg\vec{x} = f\vec{x}$. Note that $\wedge^\delta = \vee$, $\vee^\delta = \wedge$, $\leftrightarrow^\delta = +$, $\downarrow^\delta = \uparrow$, but $\neg^\delta = \neg$. In other words, $\neg$ is *self-dual.* One may check by going through all truth tables that essentially binary self-dual Boolean functions do not exist. But it was Dedekind who discovered the interesting ternary self-dual function

$$d_3 \colon (x_1, x_2, x_3) \mapsto x_1 \wedge x_2 \vee x_1 \wedge x_3 \vee x_2 \wedge x_3.$$

The above notions of duality are combined in the following

Theorem 2.4 (The duality principle for two-valued logic). *If α represents the function f then α^δ represents the dual function f^δ.*

Proof by induction on α. Trivial for $\alpha = p$. Let α, β represent f_1, f_2, respectively. Then $\alpha \wedge \beta$ represents $f : \vec{x} \mapsto f_1\vec{x} \wedge f_2\vec{x}$, and in view of the induction hypothesis, $(\alpha \wedge \beta)^\delta = \alpha^\delta \vee \beta^\delta$ represents $g : \vec{x} \mapsto f_1^\delta\vec{x} \vee f_2^\delta\vec{x}$. This function is just the dual of f because

$$f^\delta\vec{x} = \neg f \neg\vec{x} = \neg(f_1\neg\vec{x} \wedge f_2\neg\vec{x}) = \neg f_1\neg\vec{x} \vee \neg f_2\neg\vec{x} = f_1^\delta\vec{x} \vee f_2^\delta\vec{x} = g\vec{x}.$$

The induction step for $\vee$ is similar. Now let α represent f. Then $\neg\alpha$ represents $\neg f : \vec{x} \mapsto \neg f\vec{x}$. By the induction hypothesis, α^δ represents f^δ.

Thus $(\neg\alpha)^\delta = \neg\alpha^\delta$ represents $\neg f^\delta$, which coincides with $(\neg f)^\delta$ because of $(\neg f)^\delta\vec{x} = (\neg\neg f)\neg\vec{x} = \neg(\neg f\neg\vec{x}) = \neg(f^\delta\vec{x})$. ❑

For example, we know that $\leftrightarrow$ is represented by $p\wedge q \vee \neg p\wedge\neg q$. Hence, by Theorem 2.4, $+$ $(= \leftrightarrow^\delta)$ is represented by $(p\vee q)\wedge(\neg p\vee\neg q)$. More generally, if a canonical DNF α represents $f \in \boldsymbol{B}_n$, then the canonical CNF α^δ represents f^δ. Thus, if every $f \in \boldsymbol{B}_n$ is representable by a DNF then every f must necessarily be representable by a CNF, since $f \mapsto f^\delta$ maps $\boldsymbol{B}_n$ bijectively onto itself as follows from $f^{\delta^2} = f$. Note also that Dedekind's just defined ternary self-dual function d_3 shows in view of Theorem 2.4 that $p\wedge q \vee p\wedge r \vee q\wedge r \equiv (p \vee q)\wedge(p \vee r)\wedge(q \vee r)$.

Remark 2. $\{\wedge,\vee,0,1\}$ is *maximally functional incomplete*, that is, if f is any Boolean function not representable by a formula in $\wedge,\vee,0,1$, then $\{\wedge,\vee,0,1,f\}$ is functional complete (Exercise 4). As was shown by E. Post (1920), there are up to term equivalence only five maximally functional incomplete logical signatures: besides $\{\wedge,\vee,0,1\}$ only $\{\rightarrow,\wedge\}$, the dual of this, $\{\leftrightarrow,\neg\}$, and $\{d_3,\neg\}$. The formulas of the last one represent just the self-dual Boolean functions. Since $\neg p \equiv 1 + p$, the signature $\{0,1,+,\cdot\}$ is functional complete, where $\cdot$ is written in place of $\wedge$. The deeper reason is that $\{0,1,+,\cdot\}$ is at the same time the extralogical signature of fields (see **2.1**). Functional completeness in the two-valued case just derives from the fact that for a finite field, each operation on its domain is represented by a suitable polynomial. We mention also that for any finite set M of truth values considered in many-valued logics there is a generalized two-argument Sheffer function, by which every operation on M can be obtained, similarly to $\uparrow$ in the two-valued case.

Exercises

1. Verify the logical equivalences

$$\begin{aligned}(p\rightarrow q_1)\wedge(\neg p\rightarrow q_2) &\equiv p\wedge q_1 \vee \neg p\wedge q_2,\\ p_1\wedge q_1 \rightarrow p_2 \vee q_2 &\equiv (p_1\rightarrow p_2)\vee(q_1\rightarrow q_2).\end{aligned}$$

2. Show that the signatures $\{+,1\}$, $\{+,\neg\}$, $\{\leftrightarrow,0\}$, and $\{\leftrightarrow,\neg\}$ are all term equivalent. The formulas of each of these signatures represent precisely the linear Boolean functions.

3. Show that the formulas in $\wedge,\vee,0,1$ represent exactly the *monotonic* Boolean functions. These are the constants from $\boldsymbol{B}_0$, and for $n > 0$ the $f \in \boldsymbol{B}_n$ such that for all i with $1 \leqslant i \leqslant n$,

$$f(x_1,\dots,x_{i-1},0,x_{i+1},\dots,x_n) \leqslant f(x_1,\dots,x_{i-1},1,x_{i+1},\dots,x_n).$$

4. Show that the logical signature $\{\wedge, \vee, 0, 1\}$ is maximally functional incomplete.

5. If one wants to prove Corollary 2.2 syntactically with the properties of $\equiv$ (page 11) one needs generalizations of the distributivity, e.g., $\bigvee_{i\leqslant n}\alpha_i \wedge \bigvee_{j\leqslant m}\beta_j \equiv \bigvee_{i\leqslant n,\, j\leqslant m}(\alpha_i \wedge \beta_j)$. Verify the latter.

1.3 Tautologies and Logical Consequence

Instead of $w\alpha = 1$ we prefer from now on to write $w \vDash \alpha$ and read this w *satisfies* α. Further, if X is a set of formulas, we write $w \vDash X$ if $w \vDash \alpha$ for all $\alpha \in X$ and say that w is a (propositional) *model for* X. A given α (resp. X) is called *satisfiable* if there is some w with $w \vDash \alpha$ (resp. $w \vDash X$). $\vDash$, called the *satisfiability relation*, evidently has the following properties:

$$w \vDash p \;\Leftrightarrow\; wp = 1 \quad (p \in PV); \qquad w \vDash \neg\alpha \;\Leftrightarrow\; w \nvDash \alpha;$$
$$w \vDash \alpha \wedge \beta \;\Leftrightarrow\; w \vDash \alpha \text{ and } w \vDash \beta; \qquad w \vDash \alpha \vee \beta \;\Leftrightarrow\; w \vDash \alpha \text{ or } w \vDash \beta.$$

One can define the satisfiability relation $w \vDash \alpha$ for a given $w\colon PV \to \{0,1\}$ also inductively on α, according to the clauses just given. This approach is particularly useful for extending the satisfiability conditions in **2.3**.

It is obvious that $w\colon PV \to \{0,1\}$ will be uniquely determined by setting down in advance for which variables $w \vDash p$ should be valid. Likewise the notation $w \vDash \alpha$ for $\alpha \in \mathcal{F}_n$ is already meaningful when w is defined only for $p_1, \ldots, p_n$. One could extend such a w to a global valuation by setting, for instance, $wp = 0$ for all unmentioned variables p.

For formulas containing other connectives the satisfaction conditions are to be formulated accordingly. For example, we expect

$$(*) \quad w \vDash \alpha \to \beta \;\Leftrightarrow\; \text{if } w \vDash \alpha \text{ then } w \vDash \beta.$$

If $\to$ is taken to be a primitive connective, $(*)$ is required. However, we defined $\to$ in such a way that $(*)$ is provable.

Definition. α is called *logically valid* or a (two-valued) *tautology*, in short $\vDash \alpha$, whenever $w \vDash \alpha$ for all valuations w. A formula not satisfiable at all, i.e. $w \nvDash \alpha$ for all w, is called a *contradiction*.

Examples. $p \vee \neg p$ is a tautology and so is $\alpha \vee \neg\alpha$ for every formula α, the so-called *law of the excluded middle* or the *tertium non datur*. On the other hand, $\alpha \wedge \neg\alpha$ and $\alpha \leftrightarrow \neg\alpha$ are always contradictions. The following tautologies in $\rightarrow$ are mentioned in many textbooks on logic. Remember our agreement about association to the right in formulas in which $\rightarrow$ repeatedly occurs.

$p \rightarrow p$	(self-implication),
$(p \rightarrow q) \rightarrow (q \rightarrow r) \rightarrow (p \rightarrow r)$	(chain rule),
$(p \rightarrow q \rightarrow r) \rightarrow (q \rightarrow p \rightarrow r)$	(exchange of premises),
$p \rightarrow q \rightarrow p$	(premise charge),
$(p \rightarrow q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow (p \rightarrow r)$	(Frege's formula),
$((p \rightarrow q) \rightarrow p) \rightarrow p$	(Peirce's formula).

It will later turn out that all tautologies in $\rightarrow$ alone are derivable (in a sense still to be explained) from the last three formulas.

Clearly, it is decidable whether a formula α is a tautology, in that one tries out the valuations of the variables of α. Unfortunately, no essentially more efficient method is known; such a method exists only for formulas of a certain form. We will have a somewhat closer look at this problem in **4.3**. Various questions such as checking the equivalence of formulas can be reduced to a decision about whether a formula is a tautology. For notice the obvious equivalence of $\alpha \equiv \beta$ and $\vDash \alpha \leftrightarrow \beta$.

Basic in propositional logic is the following

Definition. α is a *logical consequence* of X, written $X \vDash \alpha$, if $w \vDash \alpha$ for every model w of X. In short, $w \vDash X \Rightarrow w \vDash \alpha$, for all valuations w.

While we use $\vDash$ both as the symbol for logical consequence (which is a relation between sets of formulas X and formulas α) and the satisfiability property, it will always be clear from the context what $\vDash$ actually means. Evidently, α is a tautology iff $\emptyset \vDash \alpha$, so that $\vDash \alpha$ can be regarded as an abbreviation for $\emptyset \vDash \alpha$.

In this book, $X \vDash \alpha, \beta$ will always mean '$X \vDash \alpha$ and $X \vDash \beta$'. More generally, $X \vDash Y$ is always to mean '$X \vDash \beta$ for all $\beta \in Y$'. We also write throughout $\alpha_1, \dots, \alpha_n \vDash \beta$ in place of $\{\alpha_1, \dots, \alpha_n\} \vDash \beta$, and more briefly, $X, \alpha \vDash \beta$ in place of $X \cup \{\alpha\} \vDash \beta$.

Examples of logical consequence. (a) $\alpha, \beta \vDash \alpha \wedge \beta$ and $\alpha \wedge \beta \vDash \alpha, \beta$. This is evident from the truth table of $\wedge$. (b) $\alpha, \alpha \rightarrow \beta \vDash \beta$, because $1 \rightarrow x = 1 \Rightarrow x = 1$ according to the truth table of $\rightarrow$.

(c) $X \vDash \bot \;\Rightarrow\; X \vDash \alpha$ for each α. Indeed, $X \vDash \bot = p_1 \land \neg p_1$ obviously means that X is unsatisfiable (has no model), as e.g. $X = \{p_2, \neg p_2\}$.
(d) $X, \alpha \vDash \beta \;\&\; X, \neg\alpha \vDash \beta \;\Rightarrow\; X \vDash \beta$. In order to see this let $w \vDash X$. If $w \vDash \alpha$ then $X, \alpha \vDash \beta$ and hence $w \vDash \beta$, and if $w \nvDash \alpha$ (i.e., $w \vDash \neg\alpha$) then $w \vDash \beta$ clearly follows from $X, \neg\alpha \vDash \beta$. Note that (d) reflects our case distinction made in the naive metatheory while proving (d).

Example (a) could also be stated as $X \vDash \alpha, \beta \Leftrightarrow X \vDash \alpha \land \beta$. The property exemplified by (b) is called the *modus ponens* when formulated as a rule of inference, as will be done in **1.6**. Example (d) is another formulation of the often-used procedure of proof by cases: In order to conclude a sentence β from a set of premises X it suffices to show it to be a logical consequence both under an additional supposition and under its negation. This is generalized in Exercise 3.

Important are the following general and obvious properties of $\vDash$:

$$\begin{array}{lll} \text{(R)} & \alpha \in X \Rightarrow X \vDash \alpha & (\textit{reflexivity}), \\ \text{(M)} & X \vDash \alpha \;\&\; X \subseteq X' \;\Rightarrow\; X' \vDash \alpha & (\textit{monotonicity}), \\ \text{(T)} & X \vDash Y \;\&\; Y \vDash \alpha \;\Rightarrow\; X \vDash \alpha & (\textit{transitivity}). \end{array}$$

Useful for many purposes is also the closure of the logical consequence relation under substitution, which generalizes the fact that from $p \lor \neg p$ all tautologies of the form $\alpha \lor \neg\alpha$ arise from substituting α for p.

Definition. A (propositional) *substitution* is a mapping $\sigma : PV \to \mathcal{F}$ that is extended in a natural way to a mapping $\sigma : \mathcal{F} \to \mathcal{F}$ as follows:

$$(\alpha \land \beta)^\sigma = \alpha^\sigma \land \beta^\sigma, \quad (\alpha \lor \beta)^\sigma = \alpha^\sigma \lor \beta^\sigma, \quad (\neg\alpha)^\sigma = \neg\alpha^\sigma.$$

Thus, like valuations, substitutions are considered as operations on the whole of $\mathcal{F}$. For example, if $p^\sigma = \alpha$ for some fixed p and $q^\sigma = q$ otherwise, then φ^σ arises from φ by substituting α for p at all occurrences of p in φ. From $p \lor \neg p$ arises in this way the schema $\alpha \lor \neg\alpha$. For $X \subseteq \mathcal{F}$ let $X^\sigma := \{\varphi^\sigma \mid \varphi \in X\}$. The observation $\vDash \varphi \;\Rightarrow\; \vDash \varphi^\sigma$ turns out to be the special instance $X = \emptyset$ of the useful property

$$\text{(S)} \quad X \vDash \alpha \;\Rightarrow\; X^\sigma \vDash \alpha^\sigma \qquad (\textit{substitution invariance}).$$

In order to verify (S), define w^σ for a given valuation w in such a way that $w^\sigma p = wp^\sigma$. We first prove by induction on α that

$$(*) \quad w \vDash \alpha^\sigma \;\Leftrightarrow\; w^\sigma \vDash \alpha.$$

If α is prime, $(*)$ certainly holds. As regards the induction step, note that

$$\begin{aligned} w \vDash (\alpha \wedge \beta)^\sigma \Leftrightarrow w \vDash \alpha^\sigma \wedge \beta^\sigma &\Leftrightarrow w \vDash \alpha^\sigma, \beta^\sigma \\ &\Leftrightarrow w^\sigma \vDash \alpha, \beta \quad \text{(induction hypothesis)} \\ &\Leftrightarrow w^\sigma \vDash \alpha \wedge \beta. \end{aligned}$$

The reasoning for $\vee$ and $\neg$ is analogous and so $(*)$ holds. Now let $X \vDash \alpha$ and $w \vDash X^\sigma$. By $(*)$, we get $w^\sigma \vDash X$. Thus $w^\sigma \vDash \alpha$, and again by $(*)$, $w \vDash \alpha^\sigma$. This confirms (S). Another important property of $\vDash$ that is not so easily obtained will be proved in **1.4**, namely

(F) $\quad X \vDash \alpha \;\Rightarrow\; X_0 \vDash \alpha$ for some finite subset $X_0 \subseteq X$.

$\vDash$ shares the properties (R), (M), (T), and (S) with almost every classical or nonclassical propositional *consequence relation*. This is to mean a relation $\vdash$ between sets of formulas and formulas of an arbitrary propositional language $\mathcal{F}$ that has the properties corresponding to (R), (M), (T), and, as a rule, (S). These properties are the starting point for a very general theory of logical systems created by Tarski, which underpins nearly all logical systems considered in the literature. Should $\vdash$ satisfy the property corresponding to (F) then $\vdash$ is called *finitary*.

Remark 1. Sometimes (S) is not demanded in defining a consequence relation, and if (S) holds, one speaks of a *structural* consequence relation. We omit this refinement. Notions such as tautology, consistency, maximal consistency, and so on can be used with reference to *any* consequence relation $\vdash$ in an arbitrary propositional language $\mathcal{F}$. For instance, a set of formulas X is called *consistent* in $\vdash$ whenever $X \nvdash \alpha$ for some α, and *maximally consistent* if X is consistent but has no proper consistent extension. $\vdash$ itself is called consistent if $X \nvdash \alpha$ for some X and α (this is equivalent to not $\vdash \alpha$ for all α). Here as always, $\vdash \alpha$ stands for $\emptyset \vdash \alpha$. If $\mathcal{F}$ contains $\neg$ then the consistency of X is often defined by $X \vdash \alpha, \neg\alpha$ for no α. But the aforementioned definition has the advantage of being completely independent of any assumption concerning the occurring connectives. Another example of a general definition is this: A formula set X is called *deductively closed in* $\vdash$ provided $X \vdash \alpha \Rightarrow \alpha \in X$, for all $\alpha \in \mathcal{F}$. Because of (R), this condition can be replaced by $X \vdash \alpha \Leftrightarrow \alpha \in X$. Examples in $\vDash$ are the set of all tautologies and the whole of $\mathcal{F}$. The intersection of a family of deductively closed sets is again deductively closed. Hence, each $X \subseteq \mathcal{F}$ is contained in a smallest deductively closed set, called the *deductive closure* of X in $\vdash$. It equals $\{\alpha \in \mathcal{F} \mid X \vdash \alpha\}$, as is easily seen. The notion of a consequence relation can also be defined in terms of properties of the deductive closure. We mention that (F) holds not just for our relation $\vDash$ that is given by a two-valued matrix, but for the consequence relation of *any* finite logical matrix in *any* propositional language. This is stated and at once essentially generalized in Exercise 3 in **5.7** as an application of the ultraproduct theorem.

A special property of the consequence relation $\vDash$, easily provable, is

$$\text{(D)} \quad X, \alpha \vDash \beta \;\Rightarrow\; X \vDash \alpha \to \beta,$$

called the (semantic) *deduction theorem* for propositional logic. To see this suppose $X, \alpha \vDash \beta$ and let w be a model for X. If $w \vDash \alpha$ then by the supposition, $w \vDash \beta$, hence $w \vDash \alpha \to \beta$. If $w \nvDash \alpha$ then $w \vDash \alpha \to \beta$ as well. Hence $X \vDash \alpha \to \beta$ in any case. This proves (D). As is immediately seen, the converse of (D) holds as well, that is, one may replace $\Rightarrow$ in (D) by $\Leftrightarrow$. Iterated application of this simple observation yields

$$\alpha_1, \dots, \alpha_n \vDash \beta \Leftrightarrow \vDash \alpha_1 \to \alpha_2 \to \cdots \to \alpha_n \to \beta \Leftrightarrow \vDash \alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_n \to \beta.$$

In this way, β's being a logical consequence of a finite set of premises is transformed into a tautology. Using (D) it is easy to obtain tautologies. For instance, to prove $\vDash p \to q \to p$, it is enough to verify $p \vDash q \to p$, for which it in turn suffices to show that $p, q \vDash p$, and this is trivial.

Remark 2. By some simple applications of (D) each of the tautologies in the examples on page 18 can be obtained, except the formula of Peirce. As we shall see in Chapter **2**, all properties of $\vDash$ derived above and in the exercises will carry over to the consequence relation of a first-order language.

Exercises

1. Use the deduction theorem as in the text in order to prove

   (a) $\vDash (p \to q \to r) \to (p \to q) \to (p \to r)$,
   (b) $\vDash (p \to q) \to (q \to r) \to (p \to r)$.

2. Suppose that $X \vDash \alpha \to \beta$. Prove that $X \vDash (\gamma \to \alpha) \to (\gamma \to \beta)$.

3. Verify the (rule of) *disjunctive case distinction*: if $X, \alpha \vDash \gamma$ and $X, \beta \vDash \gamma$ then $X, \alpha \vee \beta \vDash \gamma$. This implication is traditionally written more suggestively as $$\frac{X, \alpha \vDash \gamma \mid X, \beta \vDash \gamma}{X, \alpha \vee \beta \vDash \gamma}.$$

4. Verify the *rules of contraposition* (notation as in Exercise 3): $$\frac{X, \alpha \vDash \beta}{X, \neg\beta \vDash \neg\alpha} \quad ; \quad \frac{X, \neg\beta \vDash \neg\alpha}{X, \alpha \vDash \beta}.$$

5. Let $\vdash$ be a consequence relation and let X be maximally consistent in $\vdash$ (see Remark 1). Show that X is deductively closed in $\vdash$.

1.4 A Calculus of Natural Deduction

We will now define a derivability relation $\vdash$ by means of a calculus operating solely with some structural rules. $\vdash$ turns out to be identical to the consequence relation $\vDash$. The calculus $\vdash$ is of the so-called Gentzen type and its rules are given with respect to pairs (X, α) of formulas X and formulas α. Another calculus for $\vDash$, of the Hilbert type, will be considered in **1.6**. In distinction to [Ge], we do not require that X be finite; our particular goals here make such a restriction dispensable. If $\vdash$ applies to the pair (X, α) then we write $X \vdash \alpha$ and say that α is *derivable* or *provable* from X (made precise below); otherwise we write $X \nvdash \alpha$.

Following [Kl1], Gentzen's name for (X, α), *Sequenz*, is translated as *sequent*. The calculus is formulated in terms of $\wedge, \neg$ and encompasses the six rules below, called the *basic rules*. How to operate with these rules will be explained afterwards. The choice of $\{\wedge, \neg\}$ as the logical signature is a matter of convenience and justified by its functional completeness. The other standard connectives are introduced by the definitions

$$\alpha \vee \beta := \neg(\neg\alpha \wedge \neg\beta),\ \alpha \to \beta := \neg(\alpha \wedge \neg\beta),\ \alpha \leftrightarrow \beta := (\alpha \to \beta) \wedge (\beta \to \alpha).$$

$\top, \bot$ are defined as on page 5. Of course, one could choose any other functional complete signature and adapt the basic rules correspondingly. But it should be observed that a complete calculus in $\neg, \wedge, \vee, \to$, say, must also include basic rules concerning $\vee$ and $\to$, which makes induction arguments on the basic rules of the calculus more lengthy.

Each of the basic rules below has certain premises and a conclusion. Only (IS) has no premises. It allows the derivation of all sequents $\alpha \vdash \alpha$. These are called the *initial* sequents, because each derivation must start with these. (MR), the *monotonicity rule*, could be weakened. It becomes even provable if all pairs (X, α) with $\alpha \in X$ are called initial sequents.

$$\begin{array}{ll}
\text{(IS)}\ \ \dfrac{}{\alpha \vdash \alpha}\ \text{(initial sequent)} & \text{(MR)}\ \ \dfrac{X \vdash \alpha}{X' \vdash \alpha}\ (X' \supseteq X), \\[3ex]
(\wedge 1)\ \ \dfrac{X \vdash \alpha, \beta}{X \vdash \alpha \wedge \beta} & (\wedge 2)\ \ \dfrac{X \vdash \alpha \wedge \beta}{X \vdash \alpha, \beta} \\[3ex]
(\neg 1)\ \ \dfrac{X \vdash \alpha, \neg\alpha}{X \vdash \beta} & (\neg 2)\ \ \dfrac{X, \alpha \vdash \beta \ \big|\ X, \neg\alpha \vdash \beta}{X \vdash \beta}
\end{array}$$

Here and in the following $X \vdash \alpha, \beta$ is to mean $X \vdash \alpha$ and $X \vdash \beta$. This convention is important, since $X \vdash \alpha, \beta$ has another meaning in Gentzen calculi that operate with pairs of sets of formulas. The rules $(\wedge 1)$ and $(\neg 1)$ actually have two premises, just like $(\neg 2)$. Note further that $(\wedge 2)$ really consists of two subrules corresponding to the conclusions $X \vdash \alpha$ and $X \vdash \beta$. In $(\neg 2)$, X, α means $X \cup \{\alpha\}$, and this abbreviated form will always be used when there is no risk of misunderstanding.

$\alpha_1, \dots, \alpha_n \vdash \beta$ stands for $\{\alpha_1, \dots, \alpha_n\} \vdash \beta$; in particular, $\alpha \vdash \beta$ for $\{\alpha\} \vdash \beta$, and $\vdash \alpha$ for $\emptyset \vdash \alpha$, just as with $\vDash$.

$X \vdash \alpha$ (read from "X is *provable* or *derivable* α") is to mean that the sequent (X, α) can be obtained after a stepwise application of the basic rules. We can make this idea of "stepwise application" of the basic rules rigorous and formally precise (intelligible to a computer, so to speak) in the following way: a *derivation* is to mean a finite sequence $(S_0; \dots; S_n)$ of sequents such that every S_i is either an initial sequent or is obtained through the application of some basic rule to preceding elements in the sequence. Thus, *from X is derivable* α if there is a derivation $(S_0; \dots; S_n)$ with $S_n = (X, \alpha)$. A simple example with the end sequent $\alpha, \beta \vdash \alpha \wedge \beta$, or minutely $(\{\alpha, \beta\}, \alpha \wedge \beta)$, is the derivation

$$(\alpha \vdash \alpha \ ; \ \alpha, \beta \vdash \alpha \ ; \ \beta \vdash \beta \ ; \ \alpha, \beta \vdash \beta \ ; \ \alpha, \beta \vdash \alpha \wedge \beta).$$

Here (MR) was applied twice, followed by an application of $(\wedge 1)$. Not shorter would be complete derivation of the sequent $(\emptyset, \top)$, i.e., a proof of $\vdash \top$. In this example both $(\neg 1)$ and $(\neg 2)$ are essentially involved.

Useful for shortening lengthy derivations is the derivation of additional rules, which will be illustrated with the examples to follow. The second example, a generalization of the first, is the often-used proof method *reductio ad absurdum*: α is proved from X by showing that the assumption $\neg\alpha$ leads to a contradiction. The other examples are given with respect to the defined $\rightarrow$-connective. Hence, for instance, the $\rightarrow$-elimination mentioned below runs in the original language $\dfrac{X \vdash \neg(\alpha \wedge \neg\beta)}{X, \alpha \vdash \beta}$.

Examples of derivable rules

$\dfrac{X, \neg\alpha \vdash \alpha}{X \vdash \alpha}$		*proof*	*applied*
($\neg$-elimination)	1	$X, \alpha \vdash \alpha$	(IS), (MR)
	2	$X, \neg\alpha \vdash \alpha$	supposition
	3	$X \vdash \alpha$	$(\neg 2)$

		proof	*applied*
$\dfrac{X, \neg\alpha \vdash \beta, \neg\beta}{X \vdash \alpha}$ (reductio ad absurdum)	1	$X, \neg\alpha \vdash \beta, \neg\beta$	supposition
	2	$X, \neg\alpha \vdash \alpha$	$(\neg 1)$
	3	$X \vdash \alpha$	$\neg$-elimination
$\dfrac{X \vdash \alpha \to \beta}{X, \alpha \vdash \beta}$ (→-elimination)	1	$X, \alpha, \neg\beta \vdash \alpha, \neg\beta$	(IS), (MR)
	2	$X, \alpha, \neg\beta \vdash \alpha \wedge \neg\beta$	$(\wedge 1)$
	3	$X \vdash \neg(\alpha \wedge \neg\beta)$	supposition
	4	$X, \alpha, \neg\beta \vdash \neg(\alpha \wedge \neg\beta)$	(MR)
	5	$X, \alpha, \neg\beta \vdash \beta$	$(\neg 1)$ on 2 and 4
	6	$X, \alpha \vdash \beta$	$\neg$-elimination
$\dfrac{X \vdash \alpha \mid X, \alpha \vdash \beta}{X \vdash \beta}$ (cut rule)	1	$X, \neg\alpha \vdash \alpha$	supposition, (MR)
	2	$X, \neg\alpha \vdash \neg\alpha$	(IS), (MR)
	3	$X, \neg\alpha \vdash \beta$	$(\neg 1)$
	4	$X, \alpha \vdash \beta$	supposition
	5	$X \vdash \beta$	$(\neg 2)$ on 4 and 3
$\dfrac{X, \alpha \vdash \beta}{X \vdash \alpha \to \beta}$ (→-introduction)	1	$X, \alpha \wedge \neg\beta, \alpha \vdash \beta$	supposition, (MR)
	2	$X, \alpha \wedge \neg\beta \vdash \alpha$	(IS), (MR), $(\wedge 2)$
	3	$X, \alpha \wedge \neg\beta \vdash \beta$	cut rule
	4	$X, \alpha \wedge \neg\beta \vdash \neg\beta$	(IS), (MR), $(\wedge 2)$
	5	$X, \alpha \wedge \neg\beta \vdash \alpha \to \beta$	$(\neg 1)$
	6	$X, \neg(\alpha \wedge \neg\beta) \vdash \alpha \to \beta$	(IS), (MR)
	7	$X \vdash \alpha \to \beta$	$(\neg 2)$ on 5 and 6

Remark 1. The example of →-introduction is nothing other than the syntactic form of the deduction theorem that was semantically formulated in the previous section. The deduction theorem also holds for intuitionistic logic. However, it is not in general true for all logical systems dealing with implication, thus indicating that the deduction theorem is not an inherent property of every meaningful conception of implication. For instance, the deduction theorem does not hold for certain formal systems of relevance logic that attempt to model implication as a cause-and-effect relation.

A simple application of the →-elimination and the cut rule is a proof of the *detachment rule*

$$\frac{X \vdash \alpha, \alpha \to \beta}{X \vdash \beta}.$$

Indeed, the premise $X \vdash \alpha \to \beta$ yields $X, \alpha \vdash \beta$ by →-elimination, and since $X \vdash \alpha$, it follows $X \vdash \beta$ by the cut rule. Applying detachment on $X = \{\alpha, \alpha \to \beta\}$, we obtain $\alpha, \alpha \to \beta \vdash \beta$. This collection of sequents is known as *modus ponens*. It will be more closely considered in **1.6**.

Many properties of $\vdash$ are proved through rule induction, which we describe after introducing some convenient terminology. We identify a property $\mathcal{E}$ of sequents with the set of all pairs (X, α) to which $\mathcal{E}$ applies. In this sense the logical consequence relation $\vDash$ is the property that applies to all pairs (X, α) with $X \vDash \alpha$.

All the rules considered here are of the form

$$R: \ \frac{X_1 \vdash \alpha_1 \mid \ \cdots \ \mid X_n \vdash \alpha_n}{X \vdash \alpha}$$

and are referred to as Gentzen-style rules. We say that $\mathcal{E}$ is *closed under* R when $\mathcal{E}(X_1, \alpha_1), \ldots, \mathcal{E}(X_n, \alpha_n)$ implies $\mathcal{E}(X, \alpha)$. For a rule without premises, i.e., $n = 0$, this is just to mean $\mathcal{E}(X, \alpha)$. For instance, consider the above already mentioned property $\mathcal{E} : \ X \vDash \alpha$. This property is closed under each basic rule of $\vdash$. In detail this means

$$\alpha \vDash \alpha, \quad X \vDash \alpha \Rightarrow X' \vDash \alpha \text{ for } X' \supseteq X, \ \ X \vDash \alpha, \beta \Rightarrow X \vDash \alpha \wedge \beta, \ \text{ etc.}$$

From the latter we may conclude that $\mathcal{E}$ applies to all provable sequents; in other words, $\vdash$ is (semantically) *sound*. What we need here to verify this conclusion is the following easily justifiable

Principle of rule induction. *Let $\mathcal{E}$ $(\subseteq \mathfrak{P}\mathfrak{F} \times \mathfrak{F})$ be a property closed under all basic rules of $\vdash$. Then $X \vdash \alpha$ implies $\mathcal{E}(X, \alpha)$.*

Proof by induction on the length of a derivation of $S = (X, \alpha)$. If the length is 1, $\mathcal{E}S$ holds since S must be an initial sequent. Now let $(S_0; \ldots; S_n)$ be a derivation of the sequent $S := S_n$. By the induction hypothesis we have $\mathcal{E}S_i$ for all $i < n$. If S is an initial sequent then $\mathcal{E}S$ holds by assumption. Otherwise S has been obtained by the application of a basic rule on some of the S_i for $i < n$. But then $\mathcal{E}S$ holds, because $\mathcal{E}$ is closed under all basic rules. ❏

As already remarked, the property $X \vDash \alpha$ is closed under all basic rules. Therefore, the principle of rule induction immediately yields the *soundness* of the calculus, that is, $\vdash \subseteq \vDash$. More explicitly,

$$X \vdash \alpha \;\Rightarrow\; X \vDash \alpha, \text{ for all } X, \alpha.$$

There are several equivalent definitions of $\vdash$. A purely set-theoretic one is the following: $\vdash$ is the smallest of all relations $\subseteq \mathfrak{P}\mathcal{F} \times \mathcal{F}$ that are closed under all basic rules. $\vdash$ is equally the smallest consequence relation closed under the rules $(\wedge 1)$ through $(\neg 2)$. The equivalence proofs of such definitions are wordy but not particularly contentful. We therefore do not elaborate further, because we henceforth use only rule induction. Using rule induction one can also prove $X \vdash \alpha \Rightarrow X^\sigma \vdash \alpha^\sigma$, and in particular the following theorem, for which the soundness of $\vdash$ is irrelevant.

Theorem 4.1 (Finiteness theorem for $\vdash$). *If $X \vdash \alpha$ then there is a finite subset $X_0 \subseteq X$ with $X_0 \vdash \alpha$.*

Proof. Let $\mathcal{E}(X, \alpha)$ be the property '$X_0 \vdash \alpha$ for some finite $X_0 \subseteq X$'. We will show that $\mathcal{E}$ is closed under all basic rules. Certainly, $\mathcal{E}(X, \alpha)$ holds for $X = \{\alpha\}$, with $X_0 = X$ so that $\mathcal{E}$ is closed under (IS). If X has a finite subset X_0 such that $X_0 \vdash \alpha$, then so too does every set X' such that $X' \supseteq X$. Hence $\mathcal{E}$ is closed under (MR). Let $\mathcal{E}(X, \alpha)$, $\mathcal{E}(X, \beta)$, with, say, $X_1 \vdash \alpha$, $X_2 \vdash \beta$ for finite $X_1, X_2 \subseteq X$. Then we also have $X_0 \vdash \alpha, \beta$ for $X_0 = X_1 \cup X_2$ by (MR). Hence $X_0 \vdash \alpha \wedge \beta$ by $(\wedge 1)$. Thus $\mathcal{E}(X, \alpha \wedge \beta)$ holds, and $\mathcal{E}$ is closed under $(\wedge 1)$. Analogously one shows the same for all remaining basic rules of $\vdash$ so that rule induction can be applied. ❑

Of great significance is the notion of formal consistency. It fully determines the derivability relation, as the lemma to come shows. It will turn out that *consistent* formalizes adequately the notion *satisfiable.*

Definition. $X \subseteq \mathcal{F}$ is called *inconsistent* (in our calculus $\vdash$) if $X \vdash \alpha$ for all $\alpha \in \mathcal{F}$, and otherwise *consistent.* X is called *maximally consistent* if X is consistent but each $Y \supset X$ is inconsistent.

The inconsistency of X can be identified by the derivability of a single formula, namely $\bot$ $(= p_1 \wedge \neg p_1)$, because $X \vdash \bot$ implies $X \vdash p_1, \neg p_1$ by $(\wedge 2)$, hence $X \vdash \alpha$ for all α by $(\neg 1)$. Conversely, when X is inconsistent then in particular $X \vdash \bot$. Thus, $X \vdash \bot$ may be read as 'X is inconsistent', and $X \nvdash \bot$ as 'X is consistent'. From this it easily follows that X is maximally consistent iff either $\alpha \in X$ or $\neg\alpha \in X$ for each α. The latter is necessary, for $\alpha, \neg\alpha \notin X$ implies $X, \alpha \vdash \bot$ and $X, \neg\alpha \vdash \bot$, hence $X \vdash \bot$ by $(\neg 2)$, contradicting the consistency of X. Sufficiency is obvious. Most important is the following lemma. It confirms that derivability is reducible to consistency and the reader should have it down pat.

Lemma 4.2. *The derivability relation* $\vdash$ *has the properties*

$$\mathtt{C}^+: \quad X \vdash \alpha \ \Leftrightarrow\ X, \neg\alpha \vdash \bot, \qquad \mathtt{C}^-: \quad X \vdash \neg\alpha \ \Leftrightarrow\ X, \alpha \vdash \bot.$$

Proof. Suppose that $X \vdash \alpha$. Then clearly $X, \neg\alpha \vdash \alpha$ and since certainly $X, \neg\alpha \vdash \neg\alpha$, we have $X, \neg\alpha \vdash \beta$ for all β by $(\neg 1)$, in particular $X, \neg\alpha \vdash \bot$. Conversely, let $X, \neg\alpha \vdash \bot$ be the case, so that in particular $X, \neg\alpha \vdash \alpha$, and thus $X \vdash \alpha$ by $\neg$-elimination on page 23. Property $\mathtt{C}^-$ is proved completely analogously. ❏

The claim $\vDash \subseteq \vdash$, not yet proved, is equivalent to $X \nvdash \alpha \Rightarrow X \nvDash \alpha$, for all X and α. But so formulated it becomes apparent what needs to be done to obtain the proof. Since $X \nvdash \alpha$ is by $\mathtt{C}^+$ equivalent to the consistency of $X' := X \cup \{\neg\alpha\}$, and $X \nvDash \alpha$ to the satisfiability of X', we need only show that consistent sets are satisfiable. To this end we state the following lemma, whose proof, exceptionally, jumps ahead of matters in that it uses Zorn's lemma from **2.1** (page 46).

Lemma 4.3 (Lindenbaum's theorem). *Every consistent set* $X \subseteq \mathcal{F}$ *can be extended to a maximally consistent set* $X' \supseteq X$*.*

Proof. Let H be the set of all consistent $Y \supseteq X$, partially ordered with respect to $\subseteq$. $H \neq \emptyset$, because $X \in H$. Let $K \subseteq H$ be a chain, i.e., $Y \subseteq Z$ or $Z \subseteq Y$, for all $Y, Z \in K$. Claim: $U := \bigcup K$ is an upper bound for K. Since $Y \in K \Rightarrow Y \subseteq U$, we have to show that U is consistent. Assume that $U \vdash \bot$. Then $U_0 \vdash \bot$ for some finite $U_0 = \{\alpha_0, \dots, \alpha_n\} \subseteq U$. If, say, $\alpha_i \in Y_i \in K$, and Y is the biggest of the sets $Y_0, \dots, Y_n$, then $\alpha_i \in Y$ for all $i \leqslant n$, hence also $Y \vdash \bot$ by (MR). This contradicts $Y \in H$ and confirms the claim. By Zorn's lemma, H has a maximal element X', which is necessarily a maximally consistent extension of X. ❏

Remark 2. The advantage of this proof is that it is free of assumptions regarding the cardinality of the language, while Lindenbaum's original construction deals with countable languages $\mathcal{F}$ only and does not require Zorn's lemma, which is equivalent to the axiom of choice. Lindenbaum's argument runs as follows: Let $X_0 := X \subseteq \mathfrak{F}$ be consistent and let $\alpha_0, \alpha_1, \ldots$ be an enumeration of $\mathfrak{F}$. Put $X_{n+1} = X_n \cup \{\alpha_n\}$ if this set is consistent and $X_{n+1} = X_n$ otherwise. Then $Y = \bigcup_{n \in \omega} X_n$ is a maximally consistent extension of X, as is easily verified. Lemma 4.3 can also be shown very similar to this approach, using an ordinal enumeration of X instead of Zorn's Lemma.

Lemma 4.4. *A maximally consistent set $X \subseteq \mathfrak{F}$ has the property*

$$[\neg] \quad X \vdash \neg\alpha \;\Leftrightarrow\; X \nvdash \alpha, \textit{ for arbitrary } \alpha.$$

Proof. If $X \vdash \neg\alpha$, then $X \vdash \alpha$ cannot hold due to the consistency of X. If, on the other hand, $X \nvdash \alpha$, then $X, \neg\alpha$ is a consistent extension of X according by $\mathtt{C}^{+}$. But then $\neg\alpha \in X$, because X is maximally consistent. Consequently $X \vdash \neg\alpha$. ❑

Lemma 4.5. *A maximally consistent set X is satisfiable.*

Proof. Define w by $w \vDash p \;\Leftrightarrow\; X \vdash p$. We will show that for all α,

$$(*) \quad X \vdash \alpha \;\Leftrightarrow\; w \vDash \alpha.$$

For prime formulas this is trivial. Further,

$$\begin{aligned}
X \vdash \alpha \wedge \beta \quad &\Leftrightarrow \quad X \vdash \alpha, \beta && \text{(rules } (\wedge 1), (\wedge 2)\text{)}\\
&\Leftrightarrow \quad w \vDash \alpha, \beta && \text{(induction hypothesis)}\\
&\Leftrightarrow \quad w \vDash \alpha \wedge \beta && \text{(definition)}\\
X \vdash \neg\alpha \quad &\Leftrightarrow \quad X \nvdash \alpha && \text{(Lemma 4.4)}\\
&\Leftrightarrow \quad w \nvDash \alpha && \text{(induction hypothesis)}\\
&\Leftrightarrow \quad w \vDash \neg\alpha && \text{(definition).}
\end{aligned}$$

By $(*)$, w is a model for X, thereby completing the proof. ❑

Only the properties $[\wedge]$ $X \vdash \alpha \wedge \beta \Leftrightarrow X \vdash \alpha, \beta$ and $[\neg]$ from Lemma 4.4 are used in the simple model construction in Lemma 4.5, which reveals the requirements for propositional model construction in the base $\{\wedge, \neg\}$. Since maximally consistent sets X are deductively closed (Exercise 5 in **1.3**), these requirements may also be stated as

$$(\wedge)\ \alpha \wedge \beta \in X \Leftrightarrow \alpha, \beta \in X \quad ; \quad (\neg)\ \neg\alpha \in X \Leftrightarrow \alpha \notin X.$$

Lemma 4.3 and Lemma 4.5 confirm the equivalence of the consistency and the satisfiability of a set of formulas. From this fact we easily obtain the main result of the present section.

Theorem 4.6 (Completeness theorem). *$X \vdash \alpha \Leftrightarrow X \vDash \alpha$, for all formula sets X and formulas α.*

Proof. The direction $\Rightarrow$ is the soundness of $\vdash$. Conversely, $X \nvdash \alpha$ implies that $X, \neg\alpha$ is consistent. Let Y be a maximally consistent extension of $X, \neg\alpha$ according to Lemma 4.3. By Lemma 4.5, Y is satisfiable, hence also $X, \neg\alpha$. Therefore $X \nvDash \alpha$. ❑

An immediate consequence of Theorem 4.6 is the finiteness property (F) mentioned in **1.3**, which is almost trivial for $\vdash$ but not for $\vDash$:

Theorem 4.7 (Finiteness theorem for $\vDash$). *If $X \vDash \alpha$, then so too $X_0 \vDash \alpha$ for some finite subset X_0 of X.*

This is clear because the finiteness theorem holds for $\vdash$ (Theorem 4.1), hence also for $\vDash$. A further highly interesting consequence of the completeness theorem is

Theorem 4.8 (Propositional compactness theorem). *A set X of propositional formulas is satisfiable if each finite subset of X is satisfiable.*

This theorem holds because if X is unsatisfiable, i.e., if $X \vDash \bot$, then, by Theorem 4.7, we also know that $X_0 \vDash \bot$ for some finite $X_0 \subseteq X$, thus proving the claim indirectly. Conversely, one easily obtains Theorem 4.7 from Theorem 4.8. Neither Theorem 4.6 nor Theorem 4.8 make any assumptions regarding the cardinality of the set of variables. This fact has many useful applications, as the next section will illustrate.

Let us notice that there are direct proofs of Theorem 4.8 or appropriate reformulations that have nothing to do with a logical calculus. For example, the theorem is equivalent to

$$\bigcap\nolimits_{\alpha\in X} \mathrm{Md}\,\alpha = \emptyset \;\Rightarrow\; \bigcap\nolimits_{\alpha\in X_0} \mathrm{Md}\,\alpha = \emptyset \text{ for some finite } X_0 \subseteq X,$$

where $\mathrm{Md}\,\alpha$ denotes the set of all models of α. In this formulation the compactness of a certain naturally arising topological space is claimed. The points of this space are the valuations of the variables, hence the name “compactness theorem.” More on this can be found in [RS].

Another approach to completeness (probably the simplest one) is provided by Exercises 3 and 4. This approach makes some elegant use of substitutions, hence is called *the completeness proof by the substitution method*. This method is explained in the Solution Hints (and in more detail in [Ra3]). It yields the maximality of the derivability relation $\vdash$ (see Exercise 3), a much stronger result than its semantic completeness. This result yields not only the Theorems 4.6, 4.7, and 4.8 in one go, but also some further remarkable properties: Neither new tautologies nor new Hilbert style rules can consistently be adjoined to the calculus $\vdash$. These properties (discussed in detail, e.g., in [Ra1]) are known under the names *Post completeness* and *structural completeness* of $\vdash$, respectively.

Exercises

1. Prove using Theorem 4.7: if $X \cup \{\neg\alpha \mid \alpha \in Y\}$ is inconsistent and Y is nonempty, then there exist formulas $\alpha_0, \dots, \alpha_n \in Y$ such that $X \vdash \alpha_0 \vee \cdots \vee \alpha_n$.

2. Augment the signature $\{\neg, \wedge\}$ by $\vee$ and prove the completeness of the calculus obtained by supplementing the basic rules used so far with the rules
$$(\vee 1)\ \frac{X \vdash \alpha}{X \vdash \alpha \vee \beta, \beta \vee \alpha} \quad ; \quad (\vee 2)\ \frac{X, \alpha \vdash \gamma \,\big|\, X, \beta \vdash \gamma}{X, \alpha \vee \beta \vdash \gamma}\,.$$

3. Let $\vdash$ be a finitary consistent consequence relation in $\mathcal{F}\{\wedge, \neg\}$ with the properties $(\wedge 1)$ through $(\neg 2)$. Show that $\vdash$ is *maximal* (or *maximally consistent*). This means that each consequence relation $\vdash' \supset \vdash$ in $\mathcal{F}\{\wedge, \neg\}$ is inconsistent, i.e., $\vdash' \alpha$ for all α.

4. Show by referring to Exercise 3: there is exactly one (consistent) consequence relation in $\mathcal{F}\{\wedge, \neg\}$ satisfying $(\wedge 1)$–$(\neg 2)$. This clearly entails the completeness of $\vdash$.

1.5 Applications of the Compactness Theorem

Theorem 4.8 is very useful in carrying over certain properties of finite structures to infinite ones. This section presents some typical examples. While these could also be treated with the compactness theorem of first-order logic in **3.3**, the examples demonstrate how the consistency of certain sets of first-order sentences can also be obtained in propositional logic. This approach to consistency is also useful also for Herbrand's theorem and related results concerning logic programming.

1. Every set M can be (totally) ordered.[5]

This means that there is an irreflexive, transitive, and connex relation $<$ on M. For finite M this follows easily by induction on the number of elements of M. The claim is obvious when $M = \emptyset$ or is a singleton. Let now $M = N \cup \{a\}$ with an n-element set N and $a \notin N$, so that M has $n+1$ elements. Then we clearly get an order on M from that for N by "setting a to the end," that is, defining $x < a$ for all $x \in N$.

Now let M be any set. We consider for every pair $(a,b) \in M \times M$ a propositional variable p_{ab}. Let X be the set consisting of the formulas

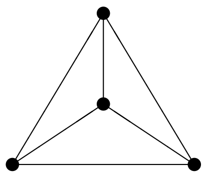
$$\begin{array}{rl} \neg p_{aa} & (a \in M), \\ p_{ab} \wedge p_{bc} \to p_{ac} & (a,b,c \in M), \\ p_{ab} \vee p_{ba} & (a \neq b). \end{array}$$

From $w \vDash X$ we obtain an order $<$, simply by putting $a < b \Leftrightarrow w \vDash p_{ab}$. $w \vDash \neg p_{aa}$ says the same thing as $a \not< a$. Analogously, the remaining formulas of X reflect transitivity and connexity. Thus, according to Theorem 4.8, it suffices to show that every finite subset $X_0 \subseteq X$ has a model. In X_0 only finitely many variables occur. Hence, there are finite sets $M_1 \subseteq M$ and $X_1 \supseteq X_0$, where X_1 is given exactly as X except that a, b, c now run through the finite set M_1 instead of M. But X_1 is satisfiable, because if $<$ orders the finite set M_1 and w is defined by $w \vDash p_{ab}$ iff $a < b$, then w is clearly a model for X_1, hence also for X_0.

2. The four-color theorem for infinite planar graphs.

A *simple graph* is a pair (V,E) with an irreflexive symmetrical relation $E \subseteq V^2$. The elements of V are called *points* or *vertices*. It is convenient to identify E with the set of all unordered pairs $\{a,b\}$ such that aEb and to call these pairs the *edges* of (V,E). If $\{a,b\} \in E$ then we say that a,b are *neighbors*. (V,E) is said to be k*-colorable* if V can be decomposed into k *color classes* $C_1, \dots, C_k \neq \emptyset$, $V = C_1 \cup \cdots \cup C_k$, with $C_i \cap C_j = \emptyset$ for $i \neq j$, such that neighboring points do not carry the same color; in other words, if $a, b \in C_i$ then $\{a,b\} \notin E$ for $i = 1, \dots, k$.

[5] Unexplained notions are defined in **2.1**. Our first application is interesting because in set theory the compactness theorem is weaker than the axiom of choice (AC) which is equivalent to the statement that every set can be well-ordered. Thus, the ordering principle is weaker than AC since it follows from the compactness theorem.



The figure shows the smallest four-colorable graph that is not three-colorable; all its points neighbor each other. We will show that a graph (V, E) is k-colorable if every finite subgraph (V_0, E_0) is k-colorable. E_0 consists of the edges $\{a, b\} \in E$ with $a, b \in V_0$. To prove our claim consider the following set X of formulas built from the variables $p_{a,i}$ for $a \in V$ and $1 \leqslant i \leqslant k$:

$$p_{a,1} \vee \cdots \vee p_{a,k}, \quad \neg(p_{a,i} \wedge p_{a,j}) \quad (a \in V,\ 1 \leqslant i < j \leqslant k),$$
$$\neg(p_{a,i} \wedge p_{b,i}) \quad (\{a, b\} \in E,\ i = 1, \ldots, k).$$

The first formula states that every point belongs to at least one color class; the second ensures their disjointedness, and the third that no neighboring points have the same color. Once again it is enough to construct some $w \vDash X$. Defining then the C_i by $a \in C_i \Leftrightarrow w \vDash p_{a,i}$ proves that (V, E) is k-colorable. We must therefore satisfy each finite $X_0 \subseteq X$. Let (V_0, E_0) be the finite subgraph of (V, E) of all the points that occur as indices in the variables of X_0. The assumption on (V_0, E_0) obviously ensures the satisfiability of X_0 for reasons analogous to those given in Example 1, and this is all we need to show. The *four-color theorem* says that every finite planar graph is four-colorable. Hence, the same holds for all graphs whose finite subgraphs are planar. These cover in particular all planar graphs embeddable in the real plane.

3. König's tree lemma. There are several versions of this lemma. For simplicity, ours refers to a *directed tree.* This is a pair $(V, \lhd)$ with an irreflexive relation $\lhd \subseteq V^2$ such that for a certain point c, the *root* of the tree, and any other point a there is precisely one *path* connecting c with a. This is a sequence $(a_i)_{i \leqslant n}$ with $a_0 = c$, $a_n = a$, and $a_i \lhd a_{i+1}$ for all $i < n$. From the uniqueness of a path connecting c with any other point it follows that each $b \neq c$ has exactly one *predecessor* in $(V, \lhd)$, that is, there is precisely one a with $a \lhd b$. Hence the name tree.

König's lemma then reads as follows: *If every $a \in V$ has only finitely many successors and V contains arbitrarily long finite paths, then there is an infinite path through V starting at c.* By such a path we mean a sequence $(c_i)_{i \in \mathbb{N}}$ such that $c_0 = c$ and $c_k \lhd c_{k+1}$ for each k. In order to prove the lemma we define the "layer" S_k inductively by $S_0 = \{c\}$ and $S_{k+1} = \{b \in V \mid \text{there is some } a \in S_k \text{ with } a \lhd b\}$. Since every point has only finitely many successors, each S_k is finite, and since there are arbitrarily long paths $c \lhd a_1 \lhd \cdots \lhd a_k$ and $a_k \in S_k$, no S_k is empty. Now let p_a for each $a \in V$ be a propositional variable, and let X consist of the formulas

$$\begin{array}{llll} \text{(A)} & \bigvee_{a\in S_k} p_a, & \neg(p_a \wedge p_b) & \big(a, b \in S_k,\ a \neq b,\ k \in \mathbb{N}\big), \\ \text{(B)} & & p_b \to p_a & \big(a, b \in V,\ a \lhd b\big). \end{array}$$

Suppose that $w \vDash X$. Then by the formulas under (A), for every k there is precisely one $a \in S_k$ with $w \vDash p_a$, denoted by c_k. In particular, $c_0 = c$. Moreover, $c_k \lhd c_{k+1}$ for all k. Indeed, if a is the predecessor of $b = c_{k+1}$, then $w \vDash p_a$ in view of (B), hence necessarily $a = c_k$. Thus, $(c_i)_{i\in\mathbb{N}}$ is a path of the type sought. Again, every finite subset $X_0 \subseteq X$ is satisfiable; for if X_0 contains variables with indices up to at most the layer S_n, then X_0 is a subset of a finite set of formulas X_1 that is defined as X, except that k runs only up to n, and for this case the claim is obvious.

4. The marriage problem (in linguistic guise).

Let $N \neq \emptyset$ be a set of *words* or *names* (in speech) with meanings in a set M. A name $\nu \in N$ can be a *synonym* (i.e., it shares its meaning with other names in N), or a *homonym* (i.e., it can have several meanings), or even both. We proceed from the plausible assumption that each name ν has finitely many meanings only and that k names have at least k meanings. It is claimed that a *pairing-off* exists; that is, an injection $f\colon N \to M$ that associates to each ν one of its original meanings.

For finite N, the claim will be proved by induction on the number n of elements of N. It is trivial for $n = 1$. Now let $n > 1$ and assume that the claim holds for all k-element sets of names whenever $0 < k < n$.

Case 1: For each k $(0 < k < n)$: k names in N have at least $k + 1$ distinct meanings. Then to an arbitrarily chosen ν from N, assign one of its meanings a to it so that from the names out of $N\setminus\{\nu\}$ any k names still have at least k meanings $\neq a$. By the induction hypothesis there is a pairing-off for $N\setminus\{\nu\}$ that together with the ordered pair (ν, a) yields a pairing-off for the whole of N.

Case 2: There is some k-element $K \subseteq N$ $(0 < k < n)$ such that the set M_K of meanings of the $\nu \in K$ has only k members. Every $\nu \in K$ can be assigned its meaning from M_K by the induction hypothesis. From the names in $N\setminus K$ any i names $(i \leqslant n - k)$ still have i meanings not in M_K, as is not hard to see. By the induction hypothesis there is also a pairing-off for $N \setminus K$ with a set of values from $M \setminus M_K$. Joining the two obviously results in a pairing-off for the whole of N.

We will now prove the claim for arbitrary sets of names N: assign to each pair $(\nu, a) \in N \times M$ a variable $p_{\nu,a}$ and consider the set of formulas

$$X: \quad \begin{cases} p_{\nu,a} \vee \cdots \vee p_{\nu,e} & (\nu \in N,\ a, \dots, e \text{ the meanings of } \nu), \\ \neg(p_{\nu,x} \wedge p_{\nu,y}) & (\nu \in N,\ x, y \in M,\ x \neq y). \end{cases}$$

Assume that $w \vDash X$. Then to each ν there is exactly one a_ν with $w \vDash p_{\nu,a_\nu}$, so that $\{(\nu, a_\nu) \mid \nu \in N\}$ is a pairing-off for N. Such a model w exists by Theorem 4.8, for in a finite set $X_0 \subseteq X$ occur only finitely many names as indices and the case of finitely many names has just been treated.

5. The ultrafilter theorem.

This theorem is of fundamental significance in topology (from which it originally stems), model theory, set theory, and elsewhere. Let I be any nonempty set. A nonempty collection of sets $F \subseteq \mathfrak{P}I$ is called a *filter on* I if for all $M, N \subseteq I$ hold the conditions

(a) $M, N \in F \Rightarrow M \cap N \in F$, (b) $M \in F \ \&\ M \subseteq N \Rightarrow N \in F$.

Since $F \neq \emptyset$, (b) shows that always $I \in F$. As is easily verified, (a) and (b) together are equivalent to just a single condition, namely to

$$(\cap) \quad M \cap N \in F \ \Leftrightarrow\ M \in F \text{ and } N \in F.$$

For fixed $K \subseteq I$, $\{J \subseteq I \mid J \supseteq K\}$ is a filter, the *principal filter* generated by K. This is a *proper filter* provided $K \neq \emptyset$, which in general is to mean a filter with $\emptyset \notin F$. Another example on an infinite I is the set of all *cofinite* subsets $M \subseteq I$, i.e., $\neg M$ $(= I \setminus M)$ is finite. This holds because $M_1 \cap M_2$ is cofinite iff M_1, M_2 are both cofinite, so that $(\cap)$ is satisfied.

A filter F is said to be an *ultrafilter on* I provided it satisfies, in addition,

$$(\neg) \quad \neg M \in F \ \Leftrightarrow\ M \notin F.$$

Ultrafilters on an infinite set I containing all cofinite subsets are called *nontrivial.* That such ultrafilters exist will be shown below. It is nearly impossible to describe them more closely. Roughly speaking, "we know they exist but we cannot see them." A trivial ultrafilter on I contains at least one finite subset. $\{J \subseteq I \mid i_0 \in J\}$ is an example for each $i_0 \in I$. This is a *principal* ultrafilter. All trivial ultrafilters are of this form, Exercise 3. Thus, trivial and principal ultrafilters coincide. In particular, each ultrafilter on a finite set I is trivial in this sense.

Each proper filter F obviously satisfies the assumption of the following theorem and can thereby be extended to an ultrafilter.

Theorem 5.1 (Ultrafilter theorem). *Every subset $F \subseteq \mathfrak{P}I$ can be extended to an ultrafilter U on a set I, provided $M_0 \cap \cdots \cap M_n \neq \emptyset$ for all n and all $M_0, \ldots, M_n \in F$.*

Proof. Consider along with the propositional variables p_J for $J \subseteq I$

$$X: \;\; p_{M \cap N} \leftrightarrow p_M \wedge p_N, \;\; p_{\neg M} \leftrightarrow \neg p_M, \;\; p_J \qquad (M, N \subseteq I, \; J \in F).$$

Let $w \vDash X$. Then $(\cap), (\neg)$ are valid for $U := \{J \subseteq I \mid w \vDash p_J\}$; hence U is an ultrafilter such that $F \subseteq U$. It therefore suffices to show that every finite subset of X has a model, for which it is in turn enough to prove the ultrafilter theorem for finite F. But this is easy: Let $F = \{M_0, \ldots, M_n\}$, $D := M_0 \cap \cdots \cap M_n$, and $i_0 \in D$. Then $U = \{J \subseteq I \mid i_0 \in J\}$ is an ultrafilter containing F. ❏

Exercises

1. Prove (using the compactness theorem) that every partial order $\leqslant_0$ on a set M can be extended to a total order $\leqslant$ on M.

2. Let F be a proper filter on I $(\neq \emptyset)$. Show that F is an ultrafilter iff it satisfies $(\cup)$: $M \cup N \in F \;\Leftrightarrow\; M \in F$ or $N \in F$.

3. Let I be an infinite set. Show that an ultrafilter U on I is trivial iff there is an $i_0 \in I$ such that $U = \{J \subseteq I \mid i_0 \in J\}$.

1.6 Hilbert Calculi

In a certain sense the simplest logical calculi are so-called *Hilbert calculi*. They are based on tautologies selected to play the role of *logical axioms*; this selection is, however, rather arbitrary and depends considerably on the logical signature. They use rules of inference such as, for example, modus ponens MP: $\alpha, \alpha \rightarrow \beta / \beta$.[6] An advantage of these calculi consists

[6] Putting it crudely, this notation should express the fact that β is held to be proved from a formula set X when α and $\alpha \rightarrow \beta$ are provable from X. Modus ponens is an example of a binary Hilbert-style rule; for a general definition of this type of rule see, for instance, [Ra1].

in the fact that formal proofs, defined below as certain finite sequences, are immediately rendered intuitive. This advantage will pay off above all in the arithmetization of proofs in **6.2**.

In the following we consider such a calculus with MP as the only rule of inference; we denote this calculus for the time being by $\mathrel{|\!\sim}$, in order to distinguish it from the calculus $\vdash$ of **1.4**. The logical signature contains just $\neg$ and $\wedge$, the same as for $\vdash$. In the axioms of $\mathrel{|\!\sim}$, however, we will also use implication defined by $\alpha \rightarrow \beta := \neg(\alpha \wedge \neg\beta)$, thus considerably shortening the writing down of the axioms.

The *logical axiom scheme* of our calculus consists of the set Λ of all formulas of the following form (not forgetting the right association of parentheses in $\Lambda 1$, $\Lambda 2$, and $\Lambda 4$):

$$\Lambda 1 \quad (\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \gamma, \qquad \Lambda 2 \quad \alpha \rightarrow \beta \rightarrow \alpha \wedge \beta,$$

$$\Lambda 3 \quad \alpha \wedge \beta \rightarrow \alpha, \ \ \alpha \wedge \beta \rightarrow \beta, \qquad \Lambda 4 \quad (\alpha \rightarrow \neg\beta) \rightarrow \beta \rightarrow \neg\alpha.$$

Λ consists only of tautologies. Moreover, all formulas derivable from Λ using MP are tautologies as well, because $\vDash \alpha, \alpha \rightarrow \beta$ implies $\vDash \beta$. We will show that *all* 2-valued tautologies are provable from Λ by means of MP. To this aim we first define the notion of a proof from $X \subseteq \mathfrak{F}$ in $\mathrel{|\!\sim}$.

Definition. A *proof* from X (in $\mathrel{|\!\sim}$) is a sequence $\Phi = (\varphi_0, \dots, \varphi_n)$ such that for every $k \leqslant n$ either $\varphi_k \in X \cup \Lambda$ or there exist indices $i, j < k$ such that $\varphi_j = \varphi_i \rightarrow \varphi_k$ (i.e., φ_k results from applying MP to terms of Φ preceding φ_k). A proof $(\varphi_0, \dots, \varphi_n)$ with $\varphi_n = \alpha$ is called *a proof of* α *from* X of length $n+1$. Whenever such a proof exists we write $X \mathrel{|\!\sim} \alpha$ and say that α is *provable* or *derivable from* X.

Example. $(p, q, p \rightarrow q \rightarrow p \wedge q, q \rightarrow p \wedge q, p \wedge q)$ is a proof of $p \wedge q$ from the set $X = \{p, q\}$. The last two terms in the proof sequence derive with MP from the previous ones, which all are members of $X \cup \Lambda$.

Since a proof contains only finitely many formulas, the preceding definition leads immediately to the finiteness theorem for $\mathrel{|\!\sim}$, formulated correspondingly to Theorem 4.1. Every proper initial segment of a proof is obviously a proof itself. Moreover, concatenating proofs of α and $\alpha \rightarrow \beta$ and tacking on β to the resulting sequence will produce a proof for β, as is plain to see. This observation implies

$$(*) \quad X \mathrel{|\!\sim} \alpha, \alpha \rightarrow \beta \ \Rightarrow\ X \mathrel{|\!\sim} \beta.$$

In short, the set of all formulas derivable from X is *closed under* MP. In applying the property $(*)$ we will often say "MP yields ..." It is easily seen that $X \mathrel{|\!\sim} \alpha$ iff α belongs to the smallest set containing $X \cup \Lambda$ and closed under MP. For the arithmetization of proofs and for automated theorem proving, however, it is more appropriate to base derivability on the notion of a proof given in the last definition, because of its finitary character. Fortunately, the following theorem relieves us of the necessity to verify a property of formulas α derivable from a given formula set X each time by induction on the length of a proof of α from X.

Theorem 6.1 (Induction principle for $\mathrel{|\!\sim}$). *Let X be given and let $\mathcal{E}$ be a property of formulas. Then $\mathcal{E}$ holds for all α with $X \mathrel{|\!\sim} \alpha$, provided*

(o) *$\mathcal{E}$ holds for all $\alpha \in X \cup \Lambda$,*

(s) *$\mathcal{E}\alpha$ and $\mathcal{E}(\alpha \to \beta)$ imply $\mathcal{E}\beta$, for all α, β.*

Proof by induction on the length n of a proof Φ of α from X. If $\alpha \in X \cup \Lambda$ then $\mathcal{E}\alpha$ holds by (o), which applies in particular if $n = 1$. If $\alpha \notin X \cup \Lambda$ then $n > 1$ and Φ contains members α_i and $\alpha_j = \alpha_i \to \alpha$ both having proofs of length $< n$. Hence, it holds $\mathcal{E}\alpha_i$ and $\mathcal{E}\alpha_j$ by the induction hypothesis, and so $\mathcal{E}\alpha$ according to (s). ❏

An application of Theorem 6.1 is the proof of $\mathrel{|\!\sim} \ \subseteq \ \vDash$, or more explicitly,

$$X \mathrel{|\!\sim} \alpha \;\Rightarrow\; X \vDash \alpha \qquad (\textit{soundness}).$$

To see this let $\mathcal{E}\alpha$ be the property '$X \vDash \alpha$' for fixed X. Certainly, $X \vDash \alpha$ holds for $\alpha \in X$. The same is true for $\alpha \in \Lambda$. Thus, $\mathcal{E}\alpha$ for all $\alpha \in X \cup \Lambda$, and (o) is confirmed. Now let $X \vDash \alpha, \alpha \to \beta$; then so too $X \vDash \beta$, thus confirming the inductive step (s) in Theorem 6.1. Consequently, $\mathcal{E}\alpha$ (that is, $X \vDash \alpha$) holds for all α with $X \mathrel{|\!\sim} \alpha$.

Unlike the proof of completeness for $\vdash$, the one for $\mathrel{|\!\sim}$ requires a whole series of derivations to be undertaken. This is in accordance with the nature of things. To get Hilbert calculi up and running one must often begin with drawn-out derivations. In the derivations below we shall use without further comment the monotonicity (M) (page 19, with $\mathrel{|\!\sim}$ for $\vDash$). (M) is obvious, for a proof in $\mathrel{|\!\sim}$ from X is also a proof from $X' \supseteq X$. Moreover, $\mathrel{|\!\sim}$ is a consequence relation (as is every Hilbert calculus, based on Hilbert style rules). For example, if $X \mathrel{|\!\sim} Y \mathrel{|\!\sim} \alpha$, we construct a proof of α from X by replacing each $\varphi \in Y$ occurring in a proof of α from Y by a proof of φ from X. This confirms the transitivity (T).

Lemma 6.2. (a) $X \mathrel{|\!\sim} \alpha \to \neg\beta \;\Rightarrow\; X \mathrel{|\!\sim} \beta \to \neg\alpha$, (b) $\mathrel{|\!\sim} \alpha \to \beta \to \alpha$,
(c) $\mathrel{|\!\sim} \alpha \to \alpha$, (d) $\mathrel{|\!\sim} \alpha \to \neg\neg\alpha$, (e) $\mathrel{|\!\sim} \beta \to \neg\beta \to \alpha$.

Proof. (a): Clearly $X \mathrel{|\!\sim} (\alpha \to \neg\beta) \to \beta \to \neg\alpha$ by Axiom Λ4. From this and from $X \mathrel{|\!\sim} \alpha \to \neg\beta$ the claim is derived by MP. (b): By Λ3, $\mathrel{|\!\sim} \beta \wedge \neg\alpha \to \neg\alpha$, and so with (a), $\mathrel{|\!\sim} \alpha \to \neg(\beta \wedge \neg\alpha) = \alpha \to \beta \to \alpha$.
(c): From $\gamma := \alpha$, $\beta := \alpha \to \alpha$ in Λ1 we obtain

$$\mathrel{|\!\sim} (\alpha \to (\alpha \to \alpha) \to \alpha) \to (\alpha \to \alpha \to \alpha) \to \alpha \to \alpha,$$

which yields the claim by applying (b) and MP twice; (d) then follows from (a) using $\mathrel{|\!\sim} \neg\alpha \to \neg\alpha$. (e): Due to $\mathrel{|\!\sim} \neg\beta \wedge \neg\alpha \to \neg\beta$ and (a), we get $\mathrel{|\!\sim} \beta \to \neg(\neg\beta \wedge \neg\alpha) = \beta \to \neg\beta \to \alpha$. ❑

Clearly, $\mathrel{|\!\sim}$ satisfies the rules ($\wedge$1) and ($\wedge$2) of **1.4**, in view of Λ2, Λ3. Part (e) of Lemma 6.2 yields $X \mathrel{|\!\sim} \beta, \neg\beta \Rightarrow X \mathrel{|\!\sim} \alpha$, so that $\mathrel{|\!\sim}$ satisfies also rule ($\neg$1). After some preparation we will show that rule ($\neg$2) holds for $\mathrel{|\!\sim}$ as well, thereby obtaining the desired completeness result. A crucial step in this direction is

Lemma 6.3 (Deduction theorem). $X, \alpha \mathrel{|\!\sim} \gamma$ *implies* $X \mathrel{|\!\sim} \alpha \to \gamma$.

Proof by induction in $\mathrel{|\!\sim}$ with a given set X, α. Let $X, \alpha \mathrel{|\!\sim} \gamma$, and let $\mathcal{E}\gamma$ now mean '$X \mathrel{|\!\sim} \alpha \to \gamma$'. To prove (o) in Theorem 6.1, let $\gamma \in \Lambda \cup X \cup \{\alpha\}$. If $\gamma = \alpha$ then clearly $X \mathrel{|\!\sim} \alpha \to \gamma$ by Lemma 6.2(c). If $\gamma \in X \cup \Lambda$ then certainly $X \mathrel{|\!\sim} \gamma$. Because also $X \mathrel{|\!\sim} \gamma \to \alpha \to \gamma$ by Lemma 6.2(b), MP yields $X \mathrel{|\!\sim} \alpha \to \gamma$, thus proving (o). To show (s) let $X, \alpha \mathrel{|\!\sim} \beta$ and $X, \alpha \mathrel{|\!\sim} \beta \to \gamma$, so that $X \mathrel{|\!\sim} \alpha \to \beta, \alpha \to \beta \to \gamma$ by the induction hypothesis. Applying MP to Λ1 twice yields $X \mathrel{|\!\sim} \alpha \to \gamma$, thus confirming (s). Therefore, by Theorem 6.1, $\mathcal{E}\gamma$ for all γ, which completes the proof. ❑

Lemma 6.4. $\mathrel{|\!\sim} \neg\neg\alpha \to \alpha$.

Proof. By Λ3 and MP, $\neg\neg\alpha \wedge \neg\alpha \mathrel{|\!\sim} \neg\alpha, \neg\neg\alpha$. Choose any τ with $\mathrel{|\!\sim} \tau$. The already verified rule ($\neg$1) clearly yields $\neg\neg\alpha \wedge \neg\alpha \mathrel{|\!\sim} \neg\tau$, and in view of Lemma 6.3, $\mathrel{|\!\sim} \neg\neg\alpha \wedge \neg\alpha \to \neg\tau$. From Lemma 6.2(a) it follows that $\mathrel{|\!\sim} \tau \to \neg(\neg\neg\alpha \wedge \neg\alpha)$. But $\mathrel{|\!\sim} \tau$, hence using MP we obtain $\mathrel{|\!\sim} \neg(\neg\neg\alpha \wedge \neg\alpha)$ and the latter formula is just $\neg\neg\alpha \to \alpha$. ❑

Lemma 6.3 and Lemma 6.4 are preparations for the next lemma, which is decisive in proving the completeness of $\mathrel{|\!\sim}$.

Lemma 6.5. $\mathrel{|\!\!\sim}$ *satisfies also rule* $(\neg 2)$ *of the calculus* $\vdash$.

Proof. Let $X, \beta \mathrel{|\!\!\sim} \alpha$ and $X, \neg\beta \mathrel{|\!\!\sim} \alpha$; then $X, \beta \mathrel{|\!\!\sim} \neg\neg\alpha$ and $X, \neg\beta \mathrel{|\!\!\sim} \neg\neg\alpha$ by Lemma 6.2(d). Hence, $X \mathrel{|\!\!\sim} \beta \to \neg\neg\alpha, \neg\beta \to \neg\neg\alpha$ (Lemma 6.3), and so $X \mathrel{|\!\!\sim} \neg\alpha \to \neg\beta$ and $X \mathrel{|\!\!\sim} \neg\alpha \to \neg\neg\beta$ by Lemma 6.2(a). Thus, MP yields $X, \neg\alpha \mathrel{|\!\!\sim} \neg\beta, \neg\neg\beta$, whence $X, \neg\alpha \mathrel{|\!\!\sim} \neg\tau$ by $(\neg 1)$, with τ as in Lemma 6.4. Therefore $X \mathrel{|\!\!\sim} \neg\alpha \to \neg\tau$, due to Lemma 6.3, and hence $X \mathrel{|\!\!\sim} \tau \to \neg\neg\alpha$ by Lemma 6.2(a). Since $X \mathrel{|\!\!\sim} \tau$ it follows that $X \mathrel{|\!\!\sim} \neg\neg\alpha$ and so eventually $X \mathrel{|\!\!\sim} \alpha$ by Lemma 6.4. ❑

Theorem 6.6 (Completeness theorem). $\mathrel{|\!\!\sim} \; = \; \vDash$.

Proof. Clearly, $\mathrel{|\!\!\sim} \; \subseteq \; \vDash$. Now, by what was said already on page 38 and by the lemma above, $\mathrel{|\!\!\sim}$ satisfies all basic rules of $\vdash$. Therefore, $\vdash \subseteq \; \mathrel{|\!\!\sim}$. Since $\vdash \; = \; \vDash$ (Theorem 4.6), we obtain also $\vDash \; \subseteq \; \mathrel{|\!\!\sim}$. ❑

This theorem implies in particular $\mathrel{|\!\!\sim} \varphi \Leftrightarrow \; \vDash \varphi$. In short, using MP one obtains from the axiom system Λ exactly the two-valued tautologies.

Remark 1. It may be something of a surprise that $\Lambda 1$–$\Lambda 4$ are sufficient to obtain all propositional tautologies, because these axioms and all formulas derivable from them using MP are collectively valid in intuitionistic and minimal logic. That Λ permits the derivation of all two-valued tautologies is based on the fact that $\to$ was *defined*. Had $\to$ been considered as a primitive connective, this would no longer have been the case. To see this, alter the interpretation of $\neg$ by setting $\neg 0 = \neg 1 = 1$. While one here indeed obtains the value 1 for every valuation of the axioms of Λ and formulas derived from them using MP, one does not do so for $\neg\neg p \to p$, which therefore cannot be derived. Modifying the two-valued matrix or using many-valued logical matrices is a widely applied method to obtain independence results for logical axioms.

Thus, we have seen that there are very different calculi for deriving tautologies or to recover other properties of the semantic relation $\vDash$. We have studied here to some extend Gentzen-style and Hilbert-style calculi and this will be done also for first-order logic in Chapter **2**. In any case, logical calculi and their completeness proofs depend essentially on the logical signature, as can be seen, for example, from Exercise 1.

Besides Gentzen- and Hilbert-style calculi there are still other types of logical calculi, for example various tableau calculi, which are above all significant for their generalizations to nonclassical logical systems. Related to tableau calculi is the resolution calculus dealt with in **4.3**.

Using Hilbert-style calculi one can axiomatize 2-valued logic in other logical signatures and functional incomplete fragments. For instance, the fragment in $\wedge,\vee$, which, while having no tautologies, contains a lot of interesting Hilbert-style rules. Proving that this fragment is axiomatizable by finitely many such rules is less easy as might be expected. At least nine Hilbert rules are required. Easier is the axiomatization of the well-known $\rightarrow$-fragment in Exercise 3, less easy that of the $\vee$-fragment in Exercise 4. Each of the infinitely many fragments of two-valued logic with or without tautologies is axiomatizable by a calculus using only *finitely many* Hilbert-style rules of its respective language, as was shown in [HeR].

Remark 2. The calculus in Exercise 4 that treats the fragment in $\vee$ alone, is based solely on unary rules. This fact considerably simplifies the matter, but the completeness proof is nevertheless nontrivial. For instance, the indispensable rule $(\alpha\beta)\gamma/\alpha(\beta\gamma)$ is derivable in this calculus, since a tricky application of the rules (3) and (4) yields $(\alpha\beta)\gamma \vdash \gamma(\alpha\beta) \vdash (\gamma\alpha)\beta \vdash \beta(\gamma\alpha) \vdash (\beta\gamma)\alpha \vdash \alpha(\beta\gamma)$. Much easier would be a completeness proof of this fragment with respect to the Gentzen-style rules ($\vee$1) and ($\vee$2) from Exercise 2 in **1.4**.

Exercises

1. Prove the completeness of the Hilbert calculus $\vdash$ in $\mathcal{F}\{\rightarrow,\bot\}$ with MP as the sole rule of inference, the definition $\neg\alpha := \alpha \rightarrow \bot$, and the axioms A1: $\alpha \rightarrow \beta \rightarrow \alpha$, A2: $(\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \gamma$, and A3: $\neg\neg\alpha \rightarrow \alpha$.

2. Let $\vdash$ be a finitary consequence relation and let $X \nvdash \varphi$. Use Zorn's lemma to prove that there is a *φ-maximal* $Y \supseteq X$, that is, $Y \nvdash \varphi$ but $Y, \alpha \vdash \varphi$ whenever $\alpha \notin Y$. Such a Y is deductively closed but need not be maximally consistent.

3. Let $\vdash$ denote the calculus in $\mathcal{F}\{\rightarrow\}$ with the rule of inference MP, the axioms A1, A2 from Exercise 1, and $((\alpha \rightarrow \beta) \rightarrow \alpha) \rightarrow \alpha$ (the Peirce axiom). Verify that (a) a φ-maximal set X is maximally consistent, (b) $\vdash$ is a complete calculus in the propositional language $\mathcal{F}\{\rightarrow\}$.

4. Show the completeness of the calculus $\vdash$ in $\mathcal{F}\{\vee\}$ with the four unary Hilbert-style rules below. The writing of $\vee$ has been omitted:

$$(1)\ \alpha/\alpha\beta, \quad (2)\ \alpha\alpha/\alpha, \quad (3)\ \alpha\beta/\beta\alpha, \quad (4)\ \alpha(\beta\gamma)/(\alpha\beta)\gamma.$$

Chapter 2

First-Order Logic

Mathematics and some other disciplines such as computer science often consider domains of individuals in which certain relations and operations are singled out. When using the language of propositional logic, our ability to talk about the properties of such relations and operations is very limited. Thus, it is necessary to refine our linguistic means of expression, in order to procure new possibilities of description. To this end, one needs not only logical symbols but also variables for the individuals of the domain being considered, as well as a symbol for equality and symbols for the relations and operations in question. First-order logic, sometimes called also predicate logic, is the part of logic that subjects properties of such relations and operations to logical analysis.

Linguistic particles such as “for all” and “there exists” (called *quantifiers*) play a central role here, whose analysis should be based on a well prepared semantic background. Hence, we first consider mathematical structures and classes of structures. Some of these are relevant both to logic (in particular model theory) and to computer science. Neither the newcomer nor the advanced student needs to read all of **2.1**, with its mathematical flavor, at once. The first five pages should suffice. The reader may continue with **2.2** and later return to what is needed.

Next we home in on the most important class of formal languages, the *first-order* languages, also called *elementary languages.* Their main characteristic is a restriction of the quantification possibilities. We discuss in detail the semantics of these languages and arrive at a notion of *logical consequence* from arbitrary premises. In this context, the notion of a formalized theory is made more precise.

Finally, we treat the introduction of new notions by explicit definitions and other expansions of a language, for instance by Skolem functions. Not until Chapter **3** do we talk about methods of formal logical deduction. While a multitude of technical details have to be considered in this chapter, nothing is especially profound. Anyway, most of it is important for the undertakings of the subsequent chapters.

2.1 Mathematical Structures

By a *structure* $\mathcal{A}$ we understand a nonempty set A together with certain distinguished relations and operations of A, as well as certain constants distinguished therein. The set A is also termed the *domain* of $\mathcal{A}$, or its *universe*. The distinguished relations, operations, and constants are called the *(basic) relations, operations, and constants of* $\mathcal{A}$. A *finite structure* is one with a finite domain. An easy example is $(\{0,1\}, \wedge, \vee, \neg)$. Here $\wedge, \vee, \neg$ have their usual meanings on the domain $\{0,1\}$, and no distinguished relations or constants occur. An *infinite structure* has an infinite domain. $\mathcal{A} = (\mathbb{N}, <, +, \cdot, 0, 1)$ is an example with the domain $\mathbb{N}$; here $<, +, \cdot, 0, 1$ have again their ordinary meaning.

Without having to say so every time, for a structure $\mathcal{A}$ the corresponding letter A will always denote the domain of $\mathcal{A}$; similarly B denotes the domain of $\mathcal{B}$, etc. If $\mathcal{A}$ contains no operations or constants, then $\mathcal{A}$ is also called a *relational structure*. If $\mathcal{A}$ has no relations it is termed an *algebraic structure*, or simply an *algebra*. For example, $(\mathbb{Z}, <)$ is a relational structure, whereas $(\mathbb{Z}, +, 0)$ is an algebraic structure, the *additive group* $\mathbb{Z}$ (it is customary to use here the symbol $\mathbb{Z}$ as well). Also the set of propositional formulas from **1.1** can be understood as an algebra, equipped with the operations $(\alpha, \beta) \mapsto (\alpha \wedge \beta)$, $(\alpha, \beta) \mapsto (\alpha \vee \beta)$, and $\alpha \mapsto \neg\alpha$. Thus, one may speak of the *formula algebra* $\mathfrak{F}$ whenever it is useful to do so.

Despite our interest in specific structures, whole classes of structures are also often considered, for instance the classes of groups, rings, fields, vector spaces, Boolean algebras, and so on. Even when initially just a single structure is viewed, call it the paradigm structure, one often needs to talk about similar structures in the same breath, in *one* language, so to speak. This can be achieved by setting aside the concrete meaning of the relation and operation symbols in the paradigm structure and considering the symbols in themselves, creating thereby a formal language that enables one to talk at once about all structures relevant to a topic. Thus, one distinguishes in this context clearly between denotation and what is denoted. To emphasize this distinction, for instance for $\mathcal{A} = (A, +, <, 0)$, it is better to write $\mathcal{A} = (A, +^{\mathcal{A}}, <^{\mathcal{A}}, 0^{\mathcal{A}})$, where $+^{\mathcal{A}}$, $<^{\mathcal{A}}$, and $0^{\mathcal{A}}$ mean the relation, operation, and constant denoted by $+$, $<$, and 0 in $\mathcal{A}$. Only if it is clear from the context what these symbols denote may the superscripts be omitted. In this way we are free to talk on the one hand about the structure $\mathcal{A}$, and on the other hand about the symbols $+, <, 0$.

A finite or infinite set L resulting in this way, consisting of relation, operation, and constant symbols of a given arity, is called an *extralogical signature*. For the class of all groups (see page 47), $L = \{\circ, e\}$ exemplifies a favored signature; that is, one often considers groups as structures of the form $(G, \circ, e)$, where $\circ$ denotes the group operation and e the unit element. But one can also define groups as structures of the signature $\{\circ\}$, because e is definable in terms of $\circ$, as we shall see later. Of course, instead of $\circ$, another operation symbol could be chosen such as $\cdot$, $*$, or $+$. The latter is mainly used in connection with commutative groups. In this sense, the actual appearance of a symbol is less important; what matters is its arity. $r \in L$ always means that r is a relation symbol, and $f \in L$ that f is an operation symbol, each time of some arity $n > 0$, which of course depends on the symbols r and f, respectively.[1]

An *L-structure* is a pair $\mathcal{A} = (A, L^{\mathcal{A}})$, where $L^{\mathcal{A}}$ contains for every $r \in L$ a relation $r^{\mathcal{A}}$ on A of the same arity as r, for every $f \in L$ an operation $f^{\mathcal{A}}$ on A of the arity of f, and for every $c \in L$ a constant $c^{\mathcal{A}} \in A$. We may omit the superscripts, provided it is clear from the context which operation or relation on A is meant. We occasionally shorten also the notation of structures. For instance, we sometimes speak of the ring $\mathbb{Z}$ or the field $\mathbb{R}$ provided there is no danger of misunderstanding.

Every structure is an L-structure for a certain signature, namely that consisting of the symbols for its relations, functions, and constants. But this does not make the name L-structure superfluous. Basic concepts, such

[1] Here r and f represent the general case and look different in a concrete situation. Relation symbols are also called predicate symbols, in particular in the unary case, and operation symbols are sometimes called function symbols. In special contexts, we also admit $n = 0$, regarding constants as 0-ary operations.

as isomorphism and substructure, refer to structures of the same signature. From **2.2** on, once the notion of a first-order language of signature L has been defined, we mostly write $\mathcal{L}$-*structure* instead of L-structure. We will then also often say that r, f, or c belongs to $\mathcal{L}$ instead of L.

If $A \subseteq B$ and f is an n-ary operation on B then A is *closed* under f, briefly f*-closed*, if $f\vec{a} \in A$ for all $\vec{a} \in A^n$. If $n = 0$, that is, if f is a constant c, this simply means $c \in A$. The intersection of any nonempty family of f-closed subsets of B is itself f-closed. Accordingly, we can talk of the smallest (the intersection) of all f-closed subsets of B that contain a given subset $E \subseteq B$. All of this extends in a natural way if f is here replaced by an arbitrary family of operations of B.

Example. For a given positive m, the set $m\mathbb{Z} := \{m \cdot n \mid n \in \mathbb{Z}\}$ of integers divisible by m is closed in $\mathbb{Z}$ under $+$, $-$, and $\cdot$, and is in fact the smallest such subset of $\mathbb{Z}$ containing m.

The *restriction* of an n-ary relation $r^B \subseteq B^n$ to a subset $A \subseteq B$ is $r^A = r^B \cap A^n$. For instance, the restriction of the standard order of $\mathbb{R}$ to $\mathbb{N}$ is the standard order of $\mathbb{N}$. Only because of this fact can the same symbol be used to denote these relations. The restriction f^A of an operation f^B on B to a set $A \subseteq B$ is defined analogously whenever A is f-closed. Simply let $f^A\vec{a} = f^B\vec{a}$ for $\vec{a} \in A^n$. For instance, addition in $\mathbb{N}$ is the restriction of addition in $\mathbb{Z}$ to $\mathbb{N}$, or addition in $\mathbb{Z}$ is an extension of this operation in $\mathbb{N}$. Again, only this state of affairs allows us to denote the two operations by the same symbol.

Let $\mathcal{B}$ be an L-structure and let $A \subseteq B$ be nonempty and closed under all operations of $\mathcal{B}$; this will be taken to include $c^{\mathcal{B}} \in A$ for constant symbols $c \in L$. To such a subset A corresponds in a natural way an L-structure $\mathcal{A} = (A, L^{\mathcal{A}})$, where $r^{\mathcal{A}}$ and $f^{\mathcal{A}}$ for $r, f \in L$ are the restrictions of $r^{\mathcal{B}}$ respectively $f^{\mathcal{B}}$ to A. Finally, let $c^{\mathcal{A}} = c^{\mathcal{B}}$ for $c \in L$. The structure $\mathcal{A}$ so defined is then called a *substructure* of $\mathcal{B}$, and $\mathcal{B}$ is called an *extension* of $\mathcal{A}$, in symbols $\mathcal{A} \subseteq \mathcal{B}$. This is a certain abuse of $\subseteq$ but it does not cause confusion, since the arguments indicate what is meant.

$\mathcal{A} \subseteq \mathcal{B}$ implies $A \subseteq B$ but not conversely, in general. For example, $\mathcal{A} = (\mathbb{N}, <, +, 0)$ is a substructure of $\mathcal{B} = (\mathbb{Z}, <, +, 0)$ since $\mathbb{N}$ is closed under addition in $\mathbb{Z}$ and 0 has the same meaning in $\mathcal{A}$ and $\mathcal{B}$. Here we dropped the superscripts for $<$, $+$, and 0 because there is no risk of misunderstanding.

A nonempty subset G of the domain B of a given L-structure $\mathcal{B}$ defines a smallest substructure $\mathcal{A}$ of $\mathcal{B}$ containing G. The domain of $\mathcal{A}$ is the smallest subset of B containing G and closed under all operations of B. $\mathcal{A}$ is called the substructure *generated from G in $\mathcal{B}$*. For instance, $3\mathbb{N}$ $(= \{3n \mid n \in \mathbb{N}\})$ is the domain of the substructure generated from $\{3\}$ in $(\mathbb{N}, +, 0)$, since $3\mathbb{N}$ contains 0 and 3, is closed under $+$, and is clearly the smallest such subset of $\mathbb{N}$. A structure $\mathcal{A}$ is called *finitely generated* if for some finite $G \subseteq A$ the substructure generated from G in $\mathcal{A}$ coincides with $\mathcal{A}$. For instance, $(\mathbb{Z}, +, -, 0)$ is finitely generated by $G = \{1\}$.

If $\mathcal{A}$ is an L-structure and $L_0 \subseteq L$ then the L_0-structure $\mathcal{A}_0$ with domain A and where $\mathsf{s}^{\mathcal{A}_0} = \mathsf{s}^{\mathcal{A}}$ for all symbols $\mathsf{s} \in L_0$ is termed the *L_0-reduct of* $\mathcal{A}$, and $\mathcal{A}$ is called an *L-expansion* of $\mathcal{A}_0$. For instance, the group $(\mathbb{Z}, +, 0)$ is the $\{+, 0\}$-reduct of the ordered ring $(\mathbb{Z}, <, +, \cdot, 0)$. The notions reduct and substructure must clearly be distinguished. A reduct of $\mathcal{A}$ has always the same domain as $\mathcal{A}$, while the domain of a substructure of $\mathcal{A}$ is as a rule a proper subset of A.

Below we list some frequently cited properties of a binary relation $\lhd$ in a set A. It is convenient to write $a \lhd b$ instead of $(a, b) \in \lhd$, and $a \ntriangleleft b$ for $(a, b) \notin \lhd$. Just as $a < b < c$ often stands for $a < b \ \& \ b < c$, we write $a \lhd b \lhd c$ for $a \lhd b \ \& \ b \lhd c$. In the listing below, 'for all a' and 'there exists an a' respectively mean 'for all $a \in A$' and 'there exists some $a \in A$'. The relation $\lhd \subseteq A^2$ is called

reflexive if $a \lhd a$ for all a,
irreflexive if $a \ntriangleleft a$ for all a,
symmetric if $a \lhd b \ \Rightarrow\ b \lhd a$, for all a, b,
antisymmetric if $a \lhd b \lhd a \ \Rightarrow\ a = b$, for all a, b,
transitive if $a \lhd b \lhd c \ \Rightarrow\ a \lhd c$, for all a, b, c,
connex if $a = b$ or $a \lhd b$ or $b \lhd a$, for all a, b.

Reflexive, transitive, and symmetric relations are also called *equivalence relations*. These are often denoted by $\sim$, $\approx$, $\equiv$, $\simeq$, or similar symbols. Such a relation generates a partition of its domain whose parts, consisting of mutually equivalent elements, are called *equivalence classes*.

We now present an overview of classes of structures to which we will later refer, mainly in Chapter **5**. Hence, for the time being, the beginner may skip the following and jump to **2.2**.

1. Graphs, partial orders, and orders. A relational structure $(A, \lhd)$ with $\lhd \subseteq A^2$ is often termed a (directed) *graph*. If $\lhd$ is irreflexive and transitive we usually write $<$ for $\lhd$ and speak of a (strict) *partial order* or a *partially ordered set*, also called a *poset* for short. If we define $x \leqslant y$ by $x < y$ or $x = y$, then $\leqslant$ is reflexive, transitive, and antisymmetric, called a *reflexive partial order* of A, the one that belongs to $<$. Call $(A, \leqslant)$ a *preorder*, if $\leqslant$ is reflexive and transitive but not necessarily antisymmetric. Preorders are frequently met in real life, e.g., x is less or equally expensive as y. If $x \sim y$ means $x \leqslant y \leqslant x$ then $\sim$ is an equivalence relation on A.

A connex partial order $\mathcal{A} = (A, <)$ is called a *total* or *linear* order, briefly termed an *order* or an *ordered* or a *strictly ordered* set. $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ are examples with respect to their standard orders. Here we follow the tradition of referring to ordered sets by their domains only.

Let U be a nonempty subset of some ordered set A such that for all $a, b \in A$, $a < b \in U \Rightarrow a \in U$. Such a U is called an *initial segment* of A. In addition, let $V := A \backslash U \neq \emptyset$. Then the pair (U, V) is called a *cut*. The cut is said to be a *gap* if U has no largest and V no smallest element. However, if U has a largest element a, and V a smallest element b, then (U, V) is called a *jump*. In this case b is called *the immediate successor* of a, and a *the immediate predecessor* of b, because there is no element from A between a and b. An infinite ordered set without gaps and jumps like $\mathbb{R}$, is said to be *continuously ordered*. Such a set is easily seen to be *densely ordered*, i.e., between any two elements lies another one.

A totally ordered subset K of a partially ordered set H is called a *chain* in H. Such a K is said to be *bounded* (to the above) if there is some $b \in H$ with $a \leqslant b$ for all $a \in K$. Call $c \in H$ *maximal* in H if no $a \in H$ exists with $a > c$. An infinite partial order need not have maximal elements, nor need all chains be bounded, as is seen by the example $(\mathbb{N}, <)$. With these notions, a basic mathematical tool can now be stated:

Zorn's lemma. *If every chain in a nonempty poset H is bounded then H has a maximal element.*

A (totally) ordered set A is *well-ordered* if every nonempty subset of A has a smallest element; equivalently, there are no infinite decreasing sequences $a_0 > a_1 > \cdots$ of elements from A. Clearly, every finite ordered set is well-ordered. The simplest example of an infinite well-ordered set is $\mathbb{N}$ together with its standard order.

2. Groupoids, semigroups, and groups. Algebras $\mathcal{A} = (A, \circ)$ with an operation $\circ\colon A^2 \to A$ are termed *groupoids.* If $\circ$ is associative then $\mathcal{A}$ is called a *semigroup*, and if $\circ$ is additionally invertible, then $\mathcal{A}$ is said to be a *group.* It is provable that a group $(G, \circ)$ in this sense contains exactly one *unit element*, that is, an element e such that $x \circ e = e \circ x = x$ for all $x \in G$, also called a *neutral element.* A well-known example is the group of bijections of a set M. If the group operation $\circ$ is commutative, we speak of a *commutative* or *abelian* group.

Here are some examples of semigroups that are not groups: (a) the set of strings over some alphabet A with respect to concatenation, the *word-semigroup* or *free semigroup generated from* A. (b) the set M^M of mappings from M to itself with respect to composition. (c) $(\mathbb{N}, +)$ and $(\mathbb{N}, \cdot)$; these two are commutative semigroups. With the exception of $(M^M, \circ)$, all mentioned examples of semigroups are *regular*, which means $x \circ z = y \circ z$ or $z \circ x = z \circ y$ implies $x = y$, for all x, y, z of the domain.

Substructures of semigroups are again semigroups. Substructures of groups are in general only semigroups, as seen from $(\mathbb{N}, +) \subseteq (\mathbb{Z}, +)$. Not so in the signature $\{\circ, e, ^{-1}\}$, where e denotes the unit element and x^{-1} the inverse of x. Here all substructures are indeed subgroups. The reason is that in $\{\circ, e, ^{-1}\}$, the group axioms can be written as universally quantified equations, where for brevity, we omit the writing of "for all x, y, z," namely as $x \circ (y \circ z) = (x \circ y) \circ z,\ \ x \circ e = x,\ \ x \circ x^{-1} = e$. These equations certainly retain their validity in the transition to substructures, and they imply $e \circ x = x \circ e = x$ and $x^{-1} \circ x = x \circ x^{-1} = e$ for all x, although $\circ$ is not a commutative operation on its domain, in general.

Ordered semigroups and groups possess along with $\circ$ some order, with respect to which $\circ$ is monotonic in both arguments, like $(\mathbb{N}, +, 0, \leqslant)$. A commutative ordered semigroup $(A, +, 0, \leqslant)$ with neutral (or zero) element 0 that at the same time is the smallest element in A, and where $x \leqslant y$ iff there is some z with $x + z = y$, is called a *domain of magnitude.* Everyday examples are the domains of *length, mass, money*, etc.

3. Rings and fields. These belong to the commonly known structures. Below we list the axioms for the theory T_F of fields in $+, \cdot, 0, 1$. A *field* is a model of T_F. A *ring* is a model of the axiom system T_R for rings that derives from T_F by dropping the axioms $\mathrm{N}^\times$, $\mathrm{C}^\times$, and $\mathrm{I}^\times$, and the constant 1 from the signature. Here are the axioms of T_F:

$$\begin{array}{rll rll}
\mathrm{N}^{+}: & x+0=x & & \mathrm{N}^{\times}: & x\cdot 1=x \\
\mathrm{C}^{+}: & x+y=y+x & & \mathrm{C}^{\times}: & x\cdot y=y\cdot x \\
\mathrm{A}^{+}: & (x+y)+z=x+(y+z) & & \mathrm{A}^{\times}: & (x\cdot y)\cdot z=x\cdot(y\cdot z) \\
\mathrm{D}: & x\cdot(y+z)=x\cdot y+x\cdot z & & \mathrm{D}': & (y+z)\cdot x=y\cdot x+z\cdot x \\
\mathrm{I}^{+}: & \forall x\exists y\, x+y=0 & & \mathrm{I}^{\times}: & 0\neq 1\wedge(\forall x\neq 0)\exists y\, x\cdot y=1
\end{array}$$

In view of $\mathrm{C}^{\times}$, axiom D' is dispensable for T_F but not for T_R. When removing I^{+} from T_R, we obtain the theory of *semirings*. A well-known example is $(\mathbb{N},+,\cdot,0)$. A commutative ring that has a unit element 1 but no *zero-divisor* (i.e., $\neg\exists x\exists y(x,y\neq 0\wedge x\cdot y=0)$ is called an *integral domain*. A typical example is $(\mathbb{Z},+,\cdot,0,1)$.

Let $\mathcal{K},\mathcal{K}'$ be any fields with $\mathcal{K}\subset\mathcal{K}'$. We call $a\in K'\backslash K$ *algebraic* or *transcendental* on $\mathcal{K}$, depending on whether a is a zero of a polynomial with coefficients in K or not. If every polynomial of degree $\geqslant 1$ with coefficients in K breaks down into linear factors, as is the case for the field of complex numbers, then $\mathcal{K}$ is called *algebraically closed*, in short, $\mathcal{K}$ is a.c. These fields will be more closely inspected in **3.3** and Chapter **5**. Each field $\mathcal{K}$ has a smallest subfield $\mathcal{P}$, called a *prime field*. One says that $\mathcal{K}$ *has characteristic* 0 or p (a prime number), depending on whether $\mathcal{P}$ is isomorphic to the field $\mathbb{Q}$ or the finite field of p elements. No other prime fields exist. It is not hard to show that $\mathcal{K}$ has the characteristic p iff the sentence $\mathtt{char}_p:\ \underbrace{1+\cdots+1}_{p}=0$ holds in $\mathcal{K}$.

Rings, fields, etc. may also be *ordered*, whereby the usual monotonicity laws are required. For example, $(\mathbb{Z},<,+,\cdot,0,1)$ is the *ordered ring* of integers and $(\mathbb{N},<,+,\cdot,0,1)$ the *ordered semiring* of natural numbers.

4. Semilattices and lattices. $\mathcal{A}=(A,\circ)$ is called a *semilattice* if $\circ$ is associative, commutative, and idempotent. An example is $(\{0,1\},\circ)$ with $\circ=\wedge$. If we define $a\leqslant b :\Leftrightarrow a\circ b=a$ then $\leqslant$ is a reflexive partial order on A. Reflexivity holds, since $a\circ a=a$. As can be easily verified, $a\circ b$ is in fact the *infimum* of a,b with respect to $\leqslant$, $a\circ b=\inf\{a,b\}$, that is, $a\circ b\leqslant a,b$, and $c\leqslant a,b\Rightarrow c\leqslant a\circ b$, for all $a,b,c\in A$.

$\mathcal{A}=(A,\cap,\cup)$ is called a *lattice* if $(A,\cap)$ and $(A,\cup)$ are both semilattices and the following so-called *absorption laws* hold: $a\cap(a\cup b)=a$ and $a\cup(a\cap b)=a$. These imply $a\cap b=a\Leftrightarrow a\cup b=b$. As above, $a\leqslant b:\Leftrightarrow a\cap b=a$ defines a partial order such that $a\cap b=\inf\{a,b\}$. In addition, one has $a \cup b = \sup\{a, b\}$ (the *supremum* of a, b), which is to mean $a, b \leqslant a \cup b$, and $a, b \leqslant c \;\Rightarrow\; a \cup b \leqslant c$, for all $a, b, c \in A$. If $\mathcal{A}$ satisfies, moreover, the *distributive laws* $a \cap (b \cup c) = (a \cap b) \cup (a \cap c)$ and $a \cup (b \cap c) = (a \cup b) \cap (a \cup c)$, then $\mathcal{A}$ is termed a *distributive* lattice. For instance, the power set $\mathfrak{P}M$ with the operations $\cap$ and $\cup$ for $\cap$ and $\cup$ respectively is a distributive lattice, as is every nonempty family of subsets of M closed under $\cap$ and $\cup$, a so-called *lattice of sets.* Another important example is $(\mathbb{N}, \mathrm{gcd}, \mathrm{lcm})$. Here $\mathrm{gcd}(a, b)$ and $\mathrm{lcm}(a, b)$ denote the greatest common divisor and the least common multiple of $a, b \in \mathbb{N}$.

5. Boolean algebras. An algebra $\mathcal{A} = (A, \cap, \cup, \neg)$ where $(A, \cap, \cup)$ is a distributive lattice and in which at least the equations

$$\neg\neg a = a, \quad \neg(a \cap b) = \neg a \cup \neg b, \quad a \cap \neg a = b \cap \neg b$$

are valid is called a *Boolean algebra.* The paradigm structure is the two-element Boolean algebra $\mathit{2} := (\{0, 1\}, \wedge, \vee, \neg)$, with $\cap, \cup$ interpreted as $\wedge, \vee$, respectively. One defines the constants 0 and 1 by $0 := a \cap \neg a$ for any $a \in A$ and $1 := \neg 0$. There are many ways to characterize Boolean algebras $\mathcal{A}$, for instance, by saying that $\mathcal{A}$ satisfies all equations valid in $\mathit{2}$. The signature can also be variously selected. For example, the signature $\wedge, \vee, \neg$ is well suited to deal algebraically with two-valued propositional logic. Terms of this signature are, up to the denotation of variables, precisely the Boolean formulas from **1.1**, and a valid logical equivalence $\alpha \equiv \beta$ corresponds to the equation $\alpha = \beta$, valid in $\mathit{2}$. Further examples of Boolean algebras are the *algebras of sets* $\mathcal{A} = (A, \cap, \cup, \neg)$. Here A consists of a nonempty system of subsets of a set I, closed under $\cap$, $\cup$, and $\neg$ (complementation in I). These are the most general examples; a famous theorem, *Stone's representation theorem*, says that each Boolean algebra is isomorphic to an algebra of sets.

6. Logical L-matrices. These are structures $\mathcal{A} = (A, L^{\mathcal{A}}, D^{\mathcal{A}})$, where L contains only operation symbols (the "logical" symbols) and D denotes a unary predicate, *the set of distinguished values* of $\mathcal{A}$. Best known is the two-valued *Boolean matrix* $\mathcal{B} = (\mathit{2}, D^{\mathcal{B}})$ with $D^{\mathcal{B}} = \{1\}$. The consequence relation $\vDash_{\mathcal{A}}$ in the propositional language $\mathcal{F}$ of signature L is defined as in the two-valued case: Let $X \subseteq \mathcal{F}$ and $\varphi \in \mathcal{F}$. Then $X \vDash_{\mathcal{A}} \varphi$ if $w\varphi \in D^{\mathcal{A}}$ for every $w\colon PV \to A$ with $wX \subseteq D^{\mathcal{A}}$ ($wX := \{w\alpha \mid \alpha \in X\}$). In words, if the values of all $\alpha \in X$ are distinguished, then so too is the value of φ.

Homomorphisms and isomorphisms. The following notions are important for both mathematical and logical investigations. Much of the material presented here will be needed in Chapter 5. In the following definition, n (>0) denotes as always the arity of f or r.

Definition. Let $\mathcal{A}, \mathcal{B}$ be L-structures and $h\colon \mathcal{A} \to \mathcal{B}$ (strictly speaking $h\colon A \to B$) a mapping such that for all $f, c, r \in L$ and $\vec{a} \in A^n$,

$$\text{(H):}\ \ hf^{\mathcal{A}}\vec{a} = f^{\mathcal{B}}h\vec{a},\ hc^{\mathcal{A}} = c^{\mathcal{B}},\ r^{\mathcal{A}}\vec{a} \Rightarrow r^{\mathcal{B}}h\vec{a} \quad \big(h\vec{a} = (ha_1, \ldots, ha_n)\big).$$

Then h is called a *homomorphism*. If the third condition in (H) is replaced by the stronger condition (S): $(\exists\vec{b}{\in}A^n)(h\vec{a}{=}h\vec{b}\ \&\ r^{\mathcal{A}}\vec{b}\,) \Leftrightarrow r^{\mathcal{B}}h\vec{a}$ [2] then h is said to be a *strong homomorphism*. For algebras, the word "strong" is clearly dispensable. An injective strong homomorphism $h\colon \mathcal{A} \to \mathcal{B}$ is called an *embedding* of $\mathcal{A}$ into $\mathcal{B}$. If, in addition, h is bijective then h is called an *isomorphism*, and in case $\mathcal{A} = \mathcal{B}$, an *automorphism*.

An embedding or isomorphism $h\colon \mathcal{A} \to \mathcal{B}$ satisfies $r^{\mathcal{A}}\vec{a} \Leftrightarrow r^{\mathcal{B}}h\vec{a}$. Indeed, since $h\vec{a}{=}h\vec{b} \Leftrightarrow \vec{a}{=}\vec{b}$, (S) yields $r^{\mathcal{B}}h\vec{a} \ \Rightarrow\ (\exists\vec{b}{\in}A^n)(\vec{a}{=}\vec{b}\ \&\ r^{\mathcal{A}}\vec{b}) \ \Rightarrow\ r^{\mathcal{A}}\vec{a}$. $\mathcal{A}, \mathcal{B}$ are said to be *isomorphic*, in symbols $\mathcal{A} \simeq \mathcal{B}$, if there is an isomorphism from $\mathcal{A}$ to $\mathcal{B}$. It is readily verified that $\simeq$ is reflexive, symmetric, and transitive, hence an equivalence relation on the class of all L-structures.

Examples 1. (a) A valuation w considered in 1.1 can be regarded as a homomorphism of the propositional formula algebra $\mathcal{F}$ into the two-element Boolean algebra $\mathit{2}$. Such a $w\colon \mathcal{F} \to \mathit{2}$ is necessarily onto.

(b) Let $\mathcal{A} = (A, *)$ be a word semigroup with the concatenation operation $*$ and $\mathcal{B}$ the additive semigroup of natural numbers, considered as L-structures for $L = \{\circ\}$ with $\circ^{\mathcal{A}} = *$ and $\circ^{\mathcal{B}} = +$. Let $\mathrm{lh}(\xi)$ denote the length of a word or string $\xi \in A$. Then $\xi \mapsto \mathrm{lh}(\xi)$ is a homomorphism since $\mathrm{lh}(\xi * \eta) = \mathrm{lh}(\xi) + \mathrm{lh}(\eta)$, for all $\xi, \eta \in A$. If $\mathcal{A}$ is generated from a single letter, lh is evidently bijective, hence an isomorphism.

(c) The mapping $a \mapsto (a, 0)$ from $\mathbb{R}$ to $\mathbb{C}$ (= set of complex numbers, understood as ordered pairs of real numbers) is a good example of an embedding of the field $\mathbb{R}$ into the field $\mathbb{C}$. Nonetheless, we are used to saying that $\mathbb{R}$ is a subfield of $\mathbb{C}$, and that $\mathbb{R}$ is a subset of $\mathbb{C}$.

[2] $(\exists\vec{b}{\in}A^n)(h\vec{a}{=}h\vec{b}\ \&\ r^{\mathcal{A}}\vec{b}\,)$ abbreviates 'there is some $\vec{b} \in A^n$ with $h\vec{a} = h\vec{b}$ and $r^{\mathcal{A}}\vec{b}$'. If $h\colon \mathcal{A} \to \mathcal{B}$ is onto (and only this case will occur in our applications) then (S) is equivalent to the more suggestive condition $r^{\mathcal{B}} = \{h\vec{a} \mid r^{\mathcal{A}}\vec{a}\}$.

(d) Let $\mathcal{A} = (\mathbb{R}, +, <)$ be the ordered additive group of real numbers and $\mathcal{B} = (\mathbb{R}_+, \cdot, <)$ the multiplicative group of positive reals. Then for any $b \in \mathbb{R}_+ \setminus \{1\}$ there is precisely one isomorphism $\eta\colon \mathcal{A} \to \mathcal{B}$ such that $\eta 1 = b$, namely $\eta\colon x \mapsto b^x$, the exponential function $\exp_b$ to the base b. It is even possible to *define* $\exp_b$ as this isomorphism, by first proving that—up to isomorphism—there is only one continuously ordered abelian group (first noticed in [Ta2] though not explicitly put into words).

(e) The algebras $\mathcal{A} = (\{0,1\}, +)$ and $\mathcal{B} = (\{0,1\}, \leftrightarrow)$ are only apparently different, but are in fact isomorphic, with the isomorphism δ where $\delta 0 = 1$, $\delta 1 = 0$. Thus, since $\mathcal{A}$ is a group, $\mathcal{B}$ is a group as well, which is not obvious at first glance. By adjoining the unary predicate $D = \{1\}$, $\mathcal{A}$ and $\mathcal{B}$ become (nonisomorphic) logical matrices. These actually define the two "dual" fragmentary two-valued logics for the connectives *either ... or ...*, and ... *if and only if* ..., which have many properties in common.

Congruences. A *congruence relation* (or simply a *congruence*) in a structure $\mathcal{A}$ of signature L is an equivalence relation $\approx$ in A such that for all $n > 0$, all $f \in L$ of arity n, and all $\vec{a}, \vec{b} \in A^n$,

$$\vec{a} \approx \vec{b} \;\Rightarrow\; f^A\vec{a} \approx f^A\vec{b}.$$

Here $\vec{a} \approx \vec{b}$ means $a_i \approx b_i$ for $i = 1, \dots, n$. A trivial example is the identity in $\mathcal{A}$. If $h\colon \mathcal{A} \to \mathcal{B}$ is a homomorphism then $\approx_h \subseteq A^2$, defined by $a \approx_h b \Leftrightarrow ha = hb$, is a congruence in $\mathcal{A}$, named the *kernel* of h. Let A' be the set of equivalence classes $a/\!\approx \;:= \{x \in A \mid a \approx x\}$ for $a \in A$, also called the *congruence classes* of $\approx$, and set $\vec{a}/\!\approx \;:= (a_1/\!\approx, \dots, a_n/\!\approx)$ for $\vec{a} \in A^n$. Define $f^{A'}(\vec{a}/\!\approx) := (f^A\vec{a})/\!\approx$ and let $r^{A'}\vec{a}/\!\approx \;:\Leftrightarrow (\exists\vec{b}{\approx}\vec{a})r^A\vec{b}$. These definitions are *sound*, that is, independent of the choice of the n-tuple $\vec{a}$ of representatives. Then A' becomes an L-structure $\mathcal{A}'$, the *factor structure of* $\mathcal{A}$ *modulo* $\approx$, denoted by $\mathcal{A}/\!\approx$. Interesting and useful, e.g. in **5.7**, is the following very general and easily provable

Homomorphism theorem. *Let $\mathcal{A}$ be an L-structure and $\approx$ a congruence in $\mathcal{A}$. Then $k\colon a \mapsto a/\!\approx$ is a strong homomorphism from $\mathcal{A}$ onto $\mathcal{A}/\!\approx$, the canonical homomorphism. Conversely, if $h\colon \mathcal{A} \to \mathcal{B}$ is a strong homomorphism from $\mathcal{A}$ onto an L-structure $\mathcal{B}$ with kernel $\approx$ then $\imath : a/\!\approx\; \mapsto ha$ is an isomorphism from $\mathcal{A}/\!\approx$ to $\mathcal{B}$, and $h = \imath \circ k$.*

Proof. We omit here the superscripts for f and r just for the sake of legibility. Clearly, $kf\vec{a} = (f\vec{a})/\!\approx \;= f(\vec{a}/\!\approx) = fk\vec{a}$ $\big(= f(ka_1, \dots, ka_n)\big)$, and $(\exists\vec{b}{\in}A^n)(k\vec{a} = k\vec{b} \ \&\ r\vec{b}) \Leftrightarrow (\exists\vec{b}{\approx}\vec{a})r\vec{b} \Leftrightarrow r\,\vec{a}/{\approx} \Leftrightarrow r\,k\vec{a}$ by definition. Hence k is what we claimed. The definition of $\imath$ is sound, and $\imath$ is bijective since $ha = hb \Rightarrow a/{\approx} = b/{\approx}$. Furthermore, $\imath$ is an isomorphism because

$$\imath f(\vec{a}/{\approx}) = hf\vec{a} = fh\vec{a} = f\imath(\vec{a}/{\approx}) \text{ and } r\,\vec{a}/{\approx} \Leftrightarrow r\,h\vec{a} \Leftrightarrow r\,\imath(\vec{a}/{\approx}).$$

Finally, h is the composition $\imath \circ k$ by the definitions of $\imath$ and k. ❑

Remark. For algebras $\mathcal{A}$, this theorem is the usual homomorphism theorem of universal algebra. $\mathcal{A}/{\approx}$ is then named the *factor algebra*. The theorem covers groups, rings, etc. In groups, the kernel of a homomorphism is already determined by the congruence class of the unit element, called a *normal subgroup*, in rings by the congruence class of 0, called an *ideal*. Hence, in textbooks on basic algebra the homomorphism theorem is separately formulated for groups and rings, but is easily derivable from the general theorem present here.

Direct products. These provide the basis for many constructions of new structures, especially in **5.7**. A well-known example is the n-dimensional vector group $(\mathbb{R}^n, 0, +)$. This is the n-fold direct product of the group $(\mathbb{R}, 0, +)$ with itself. The addition in $\mathbb{R}^n$ is defined componentwise, as is also the case in the following

Definition. Let $(\mathcal{A}_i)_{i\in I}$ be a nonempty family of L-structures. The *direct product* $\mathcal{B} = \prod_{i\in I}\mathcal{A}_i$ is the structure defined as follows: Its domain is $B = \prod_{i\in I} A_i$, called the *direct product* of the sets A_i. The elements $a = (a_i)_{i\in I}$ of B are functions defined on I with $a_i \in A_i$ for each $i \in I$. Relations and operations in $\mathcal{B}$ are defined componentwise, that is,

$$r^{\mathcal{B}}\vec{a} \ \Leftrightarrow\ r^{\mathcal{A}_i}\vec{a}_i \text{ for all } i \in I, \quad f^{\mathcal{B}}\vec{a} = (f^{\mathcal{A}_i}\vec{a}_i)_{i\in I}, \quad c^{\mathcal{B}} = (c^{\mathcal{A}_i})_{i\in I},$$

where $\vec{a} = (a^1, \dots, a^n) \in B^n$ (here the superscripts count the components) with $a^\nu := (a_i^\nu)_{i\in I}$ for $\nu = 1, \dots, n$, and $\vec{a}_i := (a_i^1, \dots, a_i^n) \in A_i^n$.

Whenever $\mathcal{A}_i = \mathcal{A}$ for all $i \in I$, then $\prod_{i\in I}\mathcal{A}_i$ is denoted by $\mathcal{A}^I$ and called a *direct power* of the structure $\mathcal{A}$. Note that $\mathcal{A}$ is embedded in $\mathcal{A}^I$ by the mapping $a \mapsto (a)_{i\in I}$, where $(a)_{i\in I}$ denotes the I-tuple with the constant value a, that is, $(a)_{i\in I} = (a, a, \dots)$. For $I = \{1, \dots, m\}$, the product $\prod_{i\in I}\mathcal{A}_i$ is also written as $\mathcal{A}_1 \times \cdots \times \mathcal{A}_m$. If $I = \{0, \dots, n{-}1\}$ one mostly writes $\mathcal{A}^n$ for $\mathcal{A}^I$.

Examples 2. (a) Let $I = \{1, 2\}$, $\mathcal{A}_i = (A_i, <^i)$, and $\mathcal{B} = \prod_{i\in I}\mathcal{A}_i$. Then $a <^{\mathcal{B}} b \Leftrightarrow a_1 <^1 b_1 \ \&\ a_2 <^2 b_2$, for all $a, b \in B = A_1 \times A_2$. Note that if $\mathcal{A}_1, \mathcal{A}_2$ are ordered sets then $\mathcal{B}$ is only a partial order. The deeper reason for this observation will become clear in Chapter **5**.

(b) Let $\mathcal{B} = \mathit{2}^I$ be a direct power of the two-element Boolean algebra $\mathit{2}$. The elements $a \in B$ are I-tuples of 0 and 1. These uniquely correspond to the subsets of I via the mapping $\imath \colon a \mapsto I_a := \{i \in I \mid a_i = 1\}$. As a matter of fact, $\imath$ is an isomorphism from $\mathcal{B}$ to $(\mathfrak{P}I, \cap, \cup, \neg)$, as can readily be verified; Exercise 4.

Exercises

1. Show that there are (up to isomorphism) exactly five two-element proper groupoids. Here a groupoid $(H, \cdot)$ is termed *proper* if the operation $\cdot$ is essentially binary.

2. $\approx$ $(\subseteq A^2)$ is termed *Euclidean* if $a \approx b \;\&\; a \approx c \;\Rightarrow\; b \approx c$, for all $a, b, c \in A$. Show that $\approx$ is an equivalence relation in A if and only if $\approx$ is reflexive and Euclidean.

3. Prove that an equivalence relation $\approx$ on an algebraic L-structure $\mathcal{A}$ is a congruence iff for all $f \in L$ of arity n, all $i = 1, \dots, n$, and all $a_1, \dots, a_{i-1}, a, a', a_{i+1}, \dots, a_n \in A$ with $a \approx a'$,
$$f(a_1, \dots, a_{i-1}, a, a_{i+1}, \dots, a_n) \approx f(a_1, \dots, a_{i-1}, a', a_{i+1}, \dots, a_n).$$

4. Prove in detail that $\mathit{2}^I \simeq (\mathfrak{P}I, \cap, \cup, \neg)$ for a nonempty index set I. Prove the corresponding statement for any subalgebra of $\mathit{2}^I$.

5. Show that $h \colon \prod_{i \in I} \mathcal{A}_i \to \mathcal{A}_j$ with $ha = a_j$ is a homomorphism for each $j \in I$.

2.2 Syntax of First-Order Languages

Standard mathematical language enables us to talk precisely about structures, such as the field of real numbers. However, for logical (and metamathematical) issues it is important to delimit the theoretical framework to be considered; this is achieved most simply by means of a formalization. In this way one obtains an *object language*; that is, the formalized elements of the language, such as the components of a structure, are *objects* of our consideration. To formalize interesting properties of a structure in this language, one requires at least variables for the elements of its domain, called *individual variables.* Further are required sufficiently many logical symbols, along with symbols for the distinguished relations, functions, and constants of the structure. These *extralogical* symbols constitute the signature L of the formal language $\mathcal{L}$ that we are going to define.

In this manner one arrives at the *first-order languages*, also termed *elementary* languages. Nothing is lost in terms of generality if the set of variables is the same for all elementary languages; we denote this set by *Var* and take it to consist of the countably many symbols $\boldsymbol{v}_0, \boldsymbol{v}_1, \dots$ Two such languages therefore differ only in the choice of their extralogical symbols. Variables for subsets of the domain are consciously excluded, since languages containing variables both for individuals and sets of these individuals—second-order languages, discussed in **3.8**—have different semantic properties from those investigated here.

We first determine the *alphabet*, the set of *basic symbols* of a first-order language $\mathcal{L}$ defined by a signature L. It includes, of course, the already specified variables $\boldsymbol{v}_0, \boldsymbol{v}_1, \dots$ In what follows, these will mostly be denoted by x, y, z, u, v, though sometimes other letters with or without indices may serve the same purpose. The boldface printed original variables are useful in writing down a formula in the variables $\boldsymbol{v}_{i_1}, \dots, \boldsymbol{v}_{i_n}$, for these can then be denoted, for instance, by $v_1, \dots, v_n$, or by $x_1, \dots, x_n$.

Further, the *logical* symbols $\wedge$ (and), $\neg$ (not), $\forall$ (*for all*), the equality sign $\boldsymbol{=}$, and, of course, all extralogical symbols from L should belong to the alphabet. Note that the boldface symbol $\boldsymbol{=}$ is taken as a basic symbol; simply taking the equality symbol $=$ could lead to unintended mix-ups with the ordinary use of $=$ (in Chapter 4 also *identity-free* languages without $\boldsymbol{=}$ will be considered). Finally, the parentheses (,) are included in the alphabet. Other symbols are introduced by definition, e.g., $\vee$, $\rightarrow$, $\leftrightarrow$ are defined as in **1.4** and the symbols $\exists$ (*there exists*) and $\exists!$ (*there exists exactly one*) will be defined later. Let $\mathcal{S}_{\mathcal{L}}$ denote the set of all strings made up of symbols that belong to the alphabet of $\mathcal{L}$.

From the set $\mathcal{S}_{\mathcal{L}}$ of all strings we pick out the meaningful ones, namely terms and formulas, according to certain rules. A term, under an interpretation of the language, will always denote an element of a domain, provided an assignment of the occurring variables to elements of that domain has been given. In order to keep the syntax as simple as possible, terms will be understood as certain parenthesis-free strings, although this kind of writing may look rather unusual at the first glance.

Terms in L:

(T1) Variables and constants, considered as atomic strings, are terms, also called *prime terms.*

(T2) If $f \in L$ is n-ary and $t_1, \dots, t_n$ are terms, then $ft_1 \cdots t_n$ is a term.

This is a recursive definition of the set of terms as a subset of $\mathcal{S}_{\mathcal{L}}$. Any string that is not generated by (T1) and (T2) is not a term in this context (cf. the related definition of $\mathcal{F}$ in **1.1**). Parenthesis-free term notation simplifies the syntax, but for binary operations we proceed differently in practice and write, for example, the term $\cdot{+}xyz$ as $(x+y) \cdot z$. The reason is that a high density of information in the notation complicates reading. Our brain does not process information sequentially like a computer. Officially, terms are parenthesis-free, and the parenthesized notation is just an alternative way of rewriting terms. Similarly to the unique reconstruction property of propositional formulas in **1.1**, here the *unique term reconstruction* property holds, that is,

$$ft_1 \cdots t_n = fs_1 \cdots s_n \textit{ implies } s_i = t_i \textit{ for } i = 1, \dots, n \quad (t_i, s_i \text{ terms}),$$

which immediately follows from the *unique term concatenation* property

$$t_1 \cdots t_n = s_1 \cdots s_m \textit{ implies } n = m \textit{ and } t_i = s_i \textit{ for } i = 1, \dots, n.$$

The latter is shown in Exercise 2. $\mathcal{T}$ $(= \mathcal{T}_L)$ denotes the set of all terms of a given signature L. The set $\mathcal{T}$ can be understood as an algebra with the operations given by $f^{\mathcal{T}}(t_1, \dots, t_n) = ft_1 \cdots t_n$, called the *term algebra.* Variable-free terms, which exist only with the availability of constant symbols, are called *constant terms* or *ground terms*, mainly in logic programming. From the definition of terms immediately follows the useful

Principle of proof by term induction. *Let $\mathcal{E}$ be a property of strings such that $\mathcal{E}$ holds for all prime terms, and for each $n > 0$ and each n-ary function symbol f, the assumptions $\mathcal{E}t_1, \dots, \mathcal{E}t_n$ imply $\mathcal{E}ft_1 \cdots t_n$. Then all terms have the property $\mathcal{E}$.*

Indeed, $\mathcal{T}$ is by definition the smallest set of strings satisfying the conditions of this principle, and hence a subset of the set of all strings with the property $\mathcal{E}$. A simple application of term induction is the proof that each compound term t is a *function term* in the sense that $t = ft_1 \cdots t_n$ for some n-ary function symbol f and some terms $t_1, \dots, t_n$. Simply consider the property 't is either prime or a function term'. Term induction can also be executed on certain subsets of $\mathcal{T}$, for instance on ground terms.

We also have at our disposal a *definition principle* by term recursion which, rather than defining it generally, we present through examples. The set $\mathit{var}\,t$ of variables occurring in a term t is recursively defined by

$$\mathit{var}\,c = \emptyset\ ;\ \ \mathit{var}\,x = \{x\}\ ;\ \ \mathit{var}\,ft_1\cdots t_n = \mathit{var}\,t_1 \cup \cdots \cup \mathit{var}\,t_n.$$

$\mathit{var}\,t$, and even $\mathit{var}\,\xi$ for any $\xi \in \mathcal{S}_{\mathcal{L}}$, can also be defined explicitly using concatenation. $\mathit{var}\,\xi$ is the set of all $x \in \mathit{Var}$ for which there are strings η, ϑ with $\xi = \eta x \vartheta$. The notion of a *subterm* of a term can also be defined recursively. Again, we can also do it more briefly using concatenation. Definition by term induction should more precisely be called *definition by term recursion.* But most authors are sloppy in this respect.

We now define recursively those strings from $\mathcal{S}_{\mathcal{L}}$ to be called *formulas*, also termed *expressions* or *well-formed formulas* of signature L.

Formulas in L:

(F1) If s, t are terms, then the string $s = t$ is a formula.

(F2) If $t_1, \dots, t_n$ are terms and $r \in L$ is n-ary, then $rt_1 \cdots t_n$ is a formula.

(F3) If α, β are formulas and x is a variable, then $(\alpha \land \beta)$, $\neg\alpha$, and $\forall x\alpha$ are formulas.

Any string not generated according to (F1), (F2), (F3) is in this context not a formula. Other logical symbols serve throughout merely as abbreviations, namely $\exists x\alpha := \neg\forall x\neg\alpha$, $(\alpha \lor \beta) := \neg(\neg\alpha \land \neg\beta)$, and as in **1.1**, $(\alpha \to \beta) := \neg(\alpha \land \neg\beta)$, and $(\alpha \leftrightarrow \beta) := ((\alpha \to \beta) \land (\beta \to \alpha))$. In addition, $s \neq t$ will throughout be written for $\neg\, s = t$. The formulas $\forall x\alpha$ and $\exists x\alpha$ are said to arise from α by *quantification.*

Examples. (a) $\forall x\exists y\, x + y = 0$ (more explicitly, $\forall x\neg\forall y\neg\, x + y = 0$) is a formula, expressing 'for all x there exists a y such that $x + y = 0$'. Here we assume tacitly that x, y denote distinct variables. The same is assumed in all of the following whenever this can be made out from the context.

(b) $\forall x\forall x\, x = y$ is a formula, since repeated quantification of the same variable is not forbidden. $\forall z\, x = y$ is a formula also if $z \neq x, y$, although z does then not appear in the formula $x = y$.

Example (b) indicates that the grammar of our formal language is more liberal than one might expect. This will spare us a lot of writing. The formulas $\forall x\forall x\, x = y$ and $\exists x\forall x\, x = y$ both have the same meaning as $\forall x\, x = y$.

These three formulas are logically equivalent (in a sense still to be defined), as are $\forall z\, x = y$ and $x = y$ for z distinct from x and y. It would be to our disadvantage to require any restriction here. In spite of this liberality, the formula syntax corresponds roughly to the syntax of natural language.

The formulas procured by (F1) and (F2) are said to be *prime* or *atomic* formulas, or simply called *prime*. As in propositional logic, prime formulas and their negations are called *literals*.

Prime formulas of the form $s = t$ are called *equations*. These are the only prime formulas if L contains no relation symbols, in which case L is called an *algebraic* signature. Prime formulas that are not equations begin with a relation symbol, although in practice a binary symbol tends to separate the two arguments as, for example, in $x \leqslant y$. The official notation is, however, that of clause (F2). The unique term concatenation property clearly implies the *unique prime formula reconstruction* property

$$rt_1 \cdots t_n = rs_1 \cdots s_n \text{ implies } t_i = s_i \text{ for } i = 1, \dots, n.$$

The set of all formulas in L is denoted by $\mathcal{L}$. The case $L = \emptyset$ defines the *language of pure identity*, denoted by $\mathcal{L}_=$, whose prime formulas are all of the form $x = y$. If $L = \{\in\}$ or $L = \{\circ\}$ then $\mathcal{L}$ is also denoted by $\mathcal{L}_\in$ or $\mathcal{L}_\circ$, resp. If L is more complex, e.g. $L = \{\circ, e\}$, we write $\mathcal{L} = \mathcal{L}\{\circ, e\}$.

Instead of terms, formulas, and structures of signature L, we will talk of $\mathcal{L}$-terms (writing $\mathcal{T}_\mathcal{L}$ for $\mathcal{T}_L$), $\mathcal{L}$-formulas, and $\mathcal{L}$-structures respectively. We also omit the prefix if $\mathcal{L}$ has been given earlier and use the same conventions of parenthesis economy as in **1.1**. We will also allow ourselves other informal aids in order to increase readability. For instance, variously shaped brackets may be used as in $\forall x \exists y \forall z[z \in y \leftrightarrow \exists u(z \in u \land u \in x)]$. Even verbal descriptions (partial or complete) are permitted, as long as the intended formula is uniquely recognizable.

The strings $\forall x$ and $\exists x$ (read "for all x" respectively "there is an x") are called *prefixes*. Also concatenations of these such as $\forall x \exists y$ are prefixes. No other prefixes are considered here. Formulas in which $\forall, \exists$ do not occur are termed *quantifier-free* or *open*. These are the Boolean combinations of prime formulas. Generally, the *Boolean combinations* of formulas from a set $X \subseteq \mathcal{L}$ are the ones generated by $\neg$, $\land$ (and $\lor$) from those of X.

X, Y, Z always denote sets of formulas, $\alpha, \beta, \gamma, \delta, \pi, \varphi, \dots$ denote formulas, and s, t terms, while Φ, Ψ are reserved to denote finite sequences of formulas and formal proofs. Substitutions (to be defined below) will be denoted by $\sigma, \tau, \omega, \rho$, and ι.

Principles of *proof by formula induction* and of *definition by formula induction* (more precisely *formula recursion*) also exist for first-order and other formal languages. After the explanation of these principles for propositional languages in **1.1**, it suffices to present here some examples, adhering to the maxim verba docent, exempla trahunt. Formula recursion is based on the *unique formula reconstruction property*, which is similar to the corresponding in **1.1**: Each composed $\varphi \in \mathcal{L}$ can uniquely be written as $\varphi = \neg\alpha$, $\varphi = (\alpha \wedge \beta)$, or $\forall x\alpha$ for some $\alpha, \beta \in \mathcal{L}$ and $x \in \mathit{Var}$. A simple example of a recursive definition is $\mathrm{rk}\,\varphi$, the *rank* of a formula φ. Starting with $\mathrm{rk}\,\pi = 0$ for prime formulas π it is defined as on page 8, with the additional clause $\mathrm{rk}\,\forall x\alpha = \mathrm{rk}\,\alpha{+}1$. Functions on $\mathcal{L}$ are sometimes defined by recursion on $\mathrm{rk}\,\varphi$, not on φ, as for instance on page 60.

Useful for some purposes is also the *quantifier rank*, $\mathrm{qr}\,\varphi$. It represents a measure of nested quantifiers in φ. For prime π let $\mathrm{qr}\,\pi = 0$, and let $\mathrm{qr}\,\neg\alpha = \mathrm{qr}\,\alpha$, $\mathrm{qr}(\alpha \wedge \beta) = \max\{\mathrm{qr}\,\alpha, \mathrm{qr}\,\beta\}$, $\mathrm{qr}\,\forall x\alpha = \mathrm{qr}\,\alpha + 1$.

Note that $\mathrm{qr}\,\exists x\varphi = \mathrm{qr}\,\neg\forall x\neg\varphi = \mathrm{qr}\,\forall x\varphi$. A *subformula* of a formula is defined analogously to the definition in **1.1**. Hence, we need say no more on this. We write $x \in \mathit{bnd}\,\varphi$ (or x occurs bound in φ) if φ contains the prefix $\forall x$. In subformulas of φ of the form $\forall x\alpha$, the formula α is called the *scope* of $\forall x$. The same prefix can occur repeatedly and with nested scopes in φ, as for instance in $\forall x(\forall x\, x{=}0 \wedge x{<}y)$. In practice we avoid this way of writing, though for a computer this would pose no problem.

Intuitively, the formulas (a) $\forall x\exists y\, x + y{=}0$ and (b) $\exists y\, x + y{=}0$ are different in that in every context with a given meaning for + and 0, the former is either true or false, whereas in (b) the variable x is waiting to be assigned a value. One also says that all variables in (a) are bound, while (b) contains the "free" variable x. The syntactic predicate 'x occurs free in φ', or '$x \in \mathit{free}\,\varphi$' is defined inductively: Let $\mathit{free}\,\alpha = \mathit{var}\,\alpha$ for prime formulas α ($\mathit{var}\,\alpha$ was defined on page 56), and

$$\mathit{free}\,(\alpha \wedge \beta) = \mathit{free}\,\alpha \cup \mathit{free}\,\beta,\ \ \mathit{free}\,\neg\alpha = \mathit{free}\,\alpha,\ \ \mathit{free}\,\forall x\alpha = \mathit{free}\,\alpha \setminus \{x\}.$$

For instance, $\mathit{free}\,(\forall x\exists y\, x{+}y{=}0) = \emptyset$, while $\mathit{free}\,(x \leqslant y \wedge \forall x\exists y\, x{+}y{=}0)$ equals $\{x, y\}$. As the last formula shows, x can occur both free and bound in a formula. This too will be avoided in practice whenever possible. In some proof-theoretically oriented presentations, even different symbols are chosen for free and bound variables. Each of these approaches has its advantages and its disadvantages.

Formulas without free variables are called *sentences*, or *closed formulas*. $1+1=0$ and $\forall x\exists y\, x+y=0$ ($=\forall x\neg\forall y\neg x+y=0$) are examples. Throughout take $\mathcal{L}^0$ to denote the set of all sentences of $\mathcal{L}$. More generally, let $\mathcal{L}^k$ be the set of all formulas φ such that $\mathit{free}\,\varphi\subseteq \mathit{Var}_k:=\{\boldsymbol{v}_0,\dots,\boldsymbol{v}_{k-1}\}$. Clearly, $\mathcal{L}^0\subseteq\mathcal{L}^1\subseteq\cdots$ and $\mathcal{L}=\bigcup_{k\in\mathbb{N}}\mathcal{L}^k$.

At this point we meet a for the remainder of the book valid

Convention. As long as not otherwise stated, the notation $\varphi=\varphi(x)$ means that the formula φ contains at most x as a free variable; more generally, $\varphi=\varphi(x_1,\dots,x_n)$ or $\varphi=\varphi(\vec{x})$ is to mean $\mathit{free}\,\varphi\subseteq\{x_1,\dots,x_n\}$, where $x_1,\dots,x_n$ stand for arbitrary but distinct variables. Not all of these variables need actually occur in φ. Further, $t=t(\vec{x})$ for terms t is to be read completely analogously.

The term $ft_1\cdots t_n$ is often denoted by $f\vec{t}$, the prime formula $rt_1\cdots t_n$ by $r\vec{t}$. Here $\vec{t}$ denotes the string concatenation $t_1\cdots t_n$. Fortunately, $\vec{t}$ behaves exactly like the sequence $(t_1,\dots,t_n)$ as was pointed out already; it has the unique term concatenation property, see page 55.

Substitutions. We begin with the substitution $\frac{t}{x}$ of some term t for a single variable x, called a *simple substitution*. Put intuitively, $\varphi\,\frac{t}{x}$ (also denoted by $\varphi_x(t)$ and read "φ t for x") is the formula that results from replacing all free occurrences of x in φ by the term t. This intuitive characterization is made precise recursively, first for terms by

$$x\,\tfrac{t}{x}=t,\quad y\,\tfrac{t}{x}=y\quad(x\neq y),\quad c\,\tfrac{t}{x}=c,\quad (ft_1\cdots t_n)\,\tfrac{t}{x}=ft_1'\cdots t_n',$$

where, for brevity, t_i' stands for $t_i\frac{t}{x}$, and next for formulas as follows:

$$\begin{array}{ll}(t_1=t_2)\tfrac{t}{x}=t_1'=t_2',\ (r\vec{t}\,)\tfrac{t}{x}=rt_1'\cdots t_n', & \multirow{2}{*}{$(\forall y\alpha)\tfrac{t}{x}=\begin{cases}\forall y\alpha & \text{if } x=y,\\ \forall y(\alpha\tfrac{t}{x}) & \text{otherwise.}\end{cases}$}\\ (\alpha\wedge\beta)\tfrac{t}{x}=\alpha\tfrac{t}{x}\wedge\beta\tfrac{t}{x},\ (\neg\alpha)\tfrac{t}{x}=\neg(\alpha\tfrac{t}{x}), & \end{array}$$

Then also $(\alpha\to\beta)\,\frac{t}{x}=\alpha\,\frac{t}{x}\to\beta\,\frac{t}{x}$, and the corresponding holds for $\vee$, while $(\exists y\alpha)\frac{t}{x}=\exists y\alpha$ for $y=x$, and $\exists y(\alpha\frac{t}{x})$ otherwise. Simple substitutions are special cases of so-called *simultaneous substitutions*

$$\varphi\,\frac{t_1\cdots t_n}{x_1\cdots x_n}\qquad(x_1,\dots,x_n\text{ distinct}).$$

For brevity, this will be written $\varphi\frac{\vec{t}}{\vec{x}}$ or $\varphi_{\vec{x}}(\vec{t}\,)$ or just $\varphi(\vec{t}\,)$, provided there is no danger of misunderstanding. Here the variables x_i are simultaneously replaced by the terms t_i at free occurrences. Simultaneous substitutions easily generalize to *global* substitutions σ. Such a σ assigns to *every* variable x a term $x^\sigma \in \mathcal{T}$. It extends to the whole of $\mathcal{T}$ by the clauses $c^\sigma = c$ and $(f\vec{t}\,)^\sigma = ft_1^\sigma \cdots t_n^\sigma$, and subsequently to $\mathcal{L}$ by recursion on $\operatorname{rk}\varphi$, so that σ is defined for the whole of $\mathcal{T} \cup \mathcal{L}$: $(t_1 = t_2)^\sigma = t_1^\sigma = t_2^\sigma$, $(r\vec{t}\,)^\sigma = rt_1^\sigma \cdots t_n^\sigma$, $(\alpha \wedge \beta)^\sigma = \alpha^\sigma \wedge \beta^\sigma$, $(\neg\alpha)^\sigma = \neg\alpha^\sigma$, and $(\forall x\varphi)^\sigma = \forall x\varphi^\tau$, where τ is defined by $x^\tau = x$ and $y^\tau = y^\sigma$ for $y \neq x$.[3]

These clauses cover also the case of a simultaneous substitution, because $\frac{\vec{t}}{\vec{x}}$ can be identified with the global substitution σ such that $x_i^\sigma = t_i$ for $i = 1, \dots, n$ and $x^\sigma = x$ otherwise. In other words, a simultaneous substitution can be understood as a global substitution σ such that $x^\sigma = x$ for *almost all* variables x, i.e., with the exception of finitely many. The *identical substitution*, always denoted by ι, is defined by $x^\iota = x$ for all x; hence $t^\iota = t$ and $\varphi^\iota = \varphi$ for all terms t and formulas φ.

Clearly, a global substitution yields *locally*, i.e. with respect to individual formulas, the same as a suitable simultaneous substitution. Moreover, it will turn out below that simultaneous substitutions are products of simple ones. Nonetheless, a separate study of simultaneous substitutions is useful mainly for Chapter 4.

It always holds that $\frac{t_1 t_2}{x_1 x_2} = \frac{t_2 t_1}{x_2 x_1}$, whereas the compositions $\frac{t_1}{x_1}\frac{t_2}{x_2}$ and $\frac{t_2}{x_2}\frac{t_1}{x_1}$ are distinct, in general. Let us elaborate by explaining the difference between $\varphi\,\frac{t_1 t_2}{x_1 x_2}$ and $\varphi\,\frac{t_1}{x_1}\frac{t_2}{x_2}$ $\big(= (\varphi\,\frac{t_1}{x_1})\,\frac{t_2}{x_2}\big)$. For example, if one wants to swap x_1, x_2 at their free occurrences in φ then the desired formula is $\varphi\,\frac{x_2 x_1}{x_1 x_2}$, but not, in general, $\varphi\,\frac{x_2}{x_1}\frac{x_1}{x_2}$ (choose for instance $\varphi = x_1 < x_2$). Rather $\varphi\,\frac{x_2 x_1}{x_1 x_2} = \varphi\,\frac{y}{x_2}\frac{x_2}{x_1}\frac{x_1}{y}$ for any $y \notin \mathit{var}\,\varphi \cup \{x_1, x_2\}$, as is readily shown by induction on φ after first treating terms. We recommend to carry out this induction in detail. In the same way we obtain

$$(1)\quad \varphi\frac{\vec{t}}{\vec{x}} = \varphi\,\frac{y}{x_n}\,\frac{t_1 \cdots t_{n\text{-}1}}{x_1 \cdots x_{n\text{-}1}}\,\frac{t_n}{y} \qquad (y \notin \mathit{var}\,\varphi \cup \mathit{var}\,\vec{x} \cup \mathit{var}\,\vec{t},\ n \geqslant 2).$$

This formula shows that a simultaneous substitution is a suitable product (composition) of simple substitutions. Conversely, it can be shown that each such product can be written as a single simultaneous substitution. In some cases (1) can be simplified. Useful, for example, is the following equation which holds in particular when all terms t_i are variable-free:

$$(2)\quad \varphi\frac{\vec{t}}{\vec{x}} = \varphi\,\frac{t_1}{x_1} \cdots \frac{t_n}{x_n} \qquad (x_i \notin \mathit{var}\,t_j \text{ for } i \neq j).$$

[3] Since $\operatorname{rk}\varphi < \operatorname{rk}\forall x\varphi$, we may assume according to the recursive construction of σ that φ^τ is already defined for all global substitutions τ.

Getting on correctly with substitutions is not altogether simple; it requires practice, because our ability to regard complex strings is not especially trustworthy. A computer is not only much faster but also more reliable in this respect.

Exercises

1. Show by term induction that a terminal segment of a term t is a concatenation $s_1 \cdots s_m$ of terms s_i for some $m \geqslant 1$. Thus, a symbol in t is at each position in t the initial symbol of a unique subterm s of t. The uniqueness of s is an easy consequence of Exercise 2(a).

2. Let $\mathcal{L}$ be a first-order language, $\mathcal{T} = \mathcal{T}_{\mathcal{L}}$, and $\mathcal{E}t$ the property 'No proper initial segment of t ($\in \mathcal{T}$) is a term, nor is t a proper initial segment of a term from $\mathcal{T}$'. Prove (a) $\mathcal{E}t$ for all $t \in \mathcal{T}$, hence $t\xi = t'\xi' \Rightarrow t = t'$ for all $t, t' \in \mathcal{T}$ and arbitrary $\xi, \xi' \in \mathcal{S}_{\mathcal{L}}$, and (b) the unique term concatenation property (page 55).

3. Prove (a) No proper initial segment of a formula φ is a formula. (b) The unique formula reconstruction property stated on page 58. (c) $\neg\xi \in \mathcal{L} \Rightarrow \xi \in \mathcal{L}$ and $\alpha, (\alpha \wedge \xi) \in \mathcal{L} \Rightarrow \xi \in \mathcal{L}$. (c) easily yields (d) $\alpha, (\alpha \rightarrow \xi) \in \mathcal{L} \Rightarrow \xi \in \mathcal{L}$, for all $\xi \in \mathcal{S}_{\mathcal{L}}$.

4. Prove that $\varphi\,\frac{t}{x} = \varphi$ for $x \notin \mathit{free}\,\varphi$, and $\varphi\,\frac{y}{x}\,\frac{t}{y} = \varphi\,\frac{t}{x}$ for $y \notin \mathit{var}\,\varphi$. It can even be shown $\varphi\,\frac{t}{x} = \varphi \Leftrightarrow x \notin \mathit{free}\,\varphi$ whenever $t \neq x$.

5. Let $X \subseteq \mathcal{L}$ be a nonempty formula set and $X^* = X \cup \{\neg\varphi \mid \varphi \in X\}$. Show that a Boolean combination of formulas from X is equivalent to a disjunction of conjunctions of formulas from X^*.

2.3 Semantics of First-Order Languages

Intuitively it is clear that the formula $\exists y\, y+y=x$ can be allocated a truth value in the domain $(\mathbb{N}, +)$ only if to the free variable x there corresponds a value in $\mathbb{N}$. Thus, along with an interpretation of the extralogical symbols, a truth value allocation for a formula φ requires a valuation of at least the variables occurring free in φ. However, it is technically more convenient to work with a global assignment of values to all variables, even if in a concrete case only the values of finitely many variables are needed. We therefore begin with the following

Definition. A *model* $\mathcal{M}$ is a pair $(\mathcal{A}, w)$ consisting of an $\mathcal{L}$-structure $\mathcal{A}$ and a *valuation* $w\colon \mathit{Var} \to A$, $w\colon x \mapsto x^w$. We denote $r^{\mathcal{A}}, f^{\mathcal{A}}, c^{\mathcal{A}}$, and x^w also by $r^{\mathcal{M}}, f^{\mathcal{M}}, c^{\mathcal{M}}$, and $x^{\mathcal{M}}$, respectively. The domain of $\mathcal{A}$ will also be called the *domain of* $\mathcal{M}$.

Models are sometimes called *interpretations*, occasionally also $\mathcal{L}$*-models* if the connection to $\mathcal{L}$ is to be highlighted. Some authors identify models with structures from the outset. This also happens in **2.5**, where we are talking about models of theories. The notion of a model is to be maintained sufficiently flexible in logic and mathematics.

A model $\mathcal{M}$ allocates in a natural way to every term t a value in A, denoted by $t^{\mathcal{M}}$ or $t^{\mathcal{A},w}$ or just by t^w. Clearly, for prime terms the value is already given by $\mathcal{M}$. This evaluation extends to compound terms by term induction as follows: $(f\vec{t}\,)^{\mathcal{M}} = f^{\mathcal{M}}\vec{t}^{\mathcal{M}}$, where $\vec{t}^{\mathcal{M}}$ abbreviates here the sequence $(t_1^{\mathcal{M}}, \dots, t_n^{\mathcal{M}})$. If the context allows we neglect the superscripts and retain just an imaginary distinction between symbols and their interpretation. For instance, if $\mathcal{A} = (\mathbb{N}, +, \cdot, 0, 1)$ and $x^w = 2$, say, we write somewhat sloppily $(0 \cdot x + 1)^{\mathcal{A},w} = 0 \cdot 2 + 1 = 1$.

The value of t under $\mathcal{M}$ depends only on the meaning of the symbols that effectively occur in t; using induction on t, the following slightly more general claim is obtained: if $\mathit{var}\, t \subseteq V \subseteq \mathit{Var}$ and $\mathcal{M}, \mathcal{M}'$ are models with the same domain such that $x^{\mathcal{M}} = x^{\mathcal{M}'}$ for all $x \in V$ and $\mathsf{s}^{\mathcal{M}} = \mathsf{s}^{\mathcal{M}'}$ for all remaining symbols s occurring in t, then $t^{\mathcal{M}} = t^{\mathcal{M}'}$. Clearly, $t^{\mathcal{A},w}$ may simply be denoted by $t^{\mathcal{A}}$, provided the term t contains no variables.

We now are going to define a satisfiability relation $\vDash$ between models $\mathcal{M} = (\mathcal{A}, w)$ and formulas φ, using induction on φ as in **1.3**. We read $\mathcal{M} \vDash \varphi$ as $\mathcal{M}$ *satisfies* φ, or $\mathcal{M}$ *is a model for* φ.

Sometimes $\mathcal{A} \vDash \varphi\,[w]$ is written instead of $\mathcal{M} \vDash \varphi$. A similar notation, just as frequently encountered, is introduced later. Each of these notations has its advantages, depending on the context. If $\mathcal{M} \vDash \varphi$ for all $\varphi \in X$ we write $\mathcal{M} \vDash X$ and call $\mathcal{M}$ a *model for* X. For the formulation of the satisfaction clauses below (taken from [Ta1]) we consider for given $\mathcal{M} = (\mathcal{A}, w)$, $x \in \mathit{Var}$, and $a \in A$ also the model $\mathcal{M}_x^a$ (generalized to $\mathcal{M}_{\vec{x}}^{\vec{a}}$

below). $\mathcal{M}_x^a$ differs from $\mathcal{M}$ only in that the variable x receives the value $a \in A$ instead of $x^{\mathcal{M}}$. Thus, $\mathcal{M}_x^a = (\mathcal{A}, w')$ with $x^{w'} = a$ and $y^{w'} = y^w$ otherwise. The satisfaction clauses then look as follows:

$$\begin{array}{lcl}
\mathcal{M} \vDash s = t & \Leftrightarrow & s^{\mathcal{M}} = t^{\mathcal{M}}, \\
\mathcal{M} \vDash r\vec{t} & \Leftrightarrow & r^{\mathcal{M}}\vec{t}^{\,\mathcal{M}}, \\
\mathcal{M} \vDash (\alpha \wedge \beta) & \Leftrightarrow & \mathcal{M} \vDash \alpha \text{ and } \mathcal{M} \vDash \beta, \\
\mathcal{M} \vDash \neg\alpha & \Leftrightarrow & \mathcal{M} \nvDash \alpha, \\
\mathcal{M} \vDash \forall x\alpha & \Leftrightarrow & \mathcal{M}_x^a \vDash \alpha \text{ for all } a \in A.
\end{array}$$

Remark 1. The last satisfaction clause can be stated differently if a name for each $a \in A$, say $\boldsymbol{a}$, is available in the signature: $\mathcal{M} \vDash \forall x\alpha \Leftrightarrow \mathcal{M} \vDash \alpha\,\frac{\boldsymbol{a}}{x}$ for all $a \in A$. This assumption permits the definition of the satisfaction relation for sentences using induction on sentences while bypassing arbitrary formulas. If not every $a \in A$ has a name in L, one could "fill up" L in advance by adjoining to L a name $\boldsymbol{a}$ for each a. But expanding the language is not always wanted and does not really simplify the matter.

$\mathcal{M}_x^a$ is slightly generalized to $\mathcal{M}_{\vec{x}}^{\vec{a}} := \mathcal{M}_{x_1 \cdots x_n}^{a_1 \cdots a_n}$ $(= (\mathcal{M}_{x_1}^{a_1})_{x_2}^{a_2} \ldots)$, which differs from $\mathcal{M}$ in the values of a sequence $x_1, \ldots, x_n$ of distinct variables. This and writing $\forall\vec{x}\varphi$ for $\forall x_1 \cdots \forall x_n\varphi$ permits a short notation of a useful generalization of the last clause above, namely

$$\mathcal{M} \vDash \forall\vec{x}\varphi \ \Leftrightarrow\ \mathcal{M}_{\vec{x}}^{\vec{a}} \vDash \varphi \text{ for all } \vec{a} \in A^n.$$

The definitions of $\alpha \vee \beta$, $\alpha \to \beta$, and $\alpha \leftrightarrow \beta$ from page 56 readily imply the additional clauses $\mathcal{M} \vDash \alpha \vee \beta$ *iff* $\mathcal{M} \vDash \alpha$ or $\mathcal{M} \vDash \beta$, $\mathcal{M} \vDash \alpha \to \beta$ *iff* $\mathcal{M} \vDash \alpha \Rightarrow \mathcal{M} \vDash \beta$, and analogously for $\leftrightarrow$. Clearly, if $\vee, \to, \leftrightarrow$ were treated as independent connectives, these equivalences would have to be added to the above ones. Further, the definition of $\exists x\varphi$ in **2.2** corresponds to its intended meaning, because $\mathcal{M} \vDash \exists x\varphi \Leftrightarrow \mathcal{M}_x^a \vDash \varphi$ for some $a \in A$. Indeed, whenever $\mathcal{M} \vDash \neg\forall x\neg\varphi$ $(= \exists x\varphi)$ then $\mathcal{M}_x^a \vDash \neg\varphi$ does not hold for all a; hence there is some $a \in A$ such that $\mathcal{M}_x^a \nvDash \neg\varphi$, or equivalently, $\mathcal{M}_x^a \vDash \varphi$. And this chain of reasoning is obviously reversible.

Example 1. $\mathcal{M} \vDash \exists x\, x = t$ for arbitrary $\mathcal{M}$, provided $x \notin var\, t$. Indeed, $\mathcal{M}_x^a \vDash x = t$ with $a := t^{\mathcal{M}}$, since $x^{\mathcal{M}_x^a} = a = t^{\mathcal{M}} = t^{\mathcal{M}_x^a}$ in view of $x \notin var\, t$. The assumption $x \notin var\, t$ is essential. For instance, $\mathcal{M} \vDash \exists x\, x = fx$ holds only if the function $f^{\mathcal{M}}$ has a fixed point.

We now introduce several fundamental notions that will be treated more systematically in **2.4** and **2.5**, once certain necessary preparations have been completed.

Definition. A formula or set of formulas in $\mathcal{L}$ is termed *satisfiable* if it has a model. $\varphi \in \mathcal{L}$ is called *generally valid, logically valid*, or a *tautology*, in short, $\vDash \varphi$, if $\mathcal{M} \vDash \varphi$ for every model $\mathcal{M}$. Formulas α, β are called (logically or semantically) *equivalent*, in symbols, $\alpha \equiv \beta$, if

$$\mathcal{M} \vDash \alpha \;\Leftrightarrow\; \mathcal{M} \vDash \beta, \text{ for each } \mathcal{L}\text{-model } \mathcal{M}.$$

Further, let $\mathcal{A} \vDash \varphi$ (read φ *holds in* $\mathcal{A}$ or $\mathcal{A}$ *satisfies* φ) if $(\mathcal{A}, w) \vDash \varphi$ for all $w\colon \mathit{Var} \to A$. One writes $\mathcal{A} \vDash X$ in case $\mathcal{A} \vDash \varphi$ for all $\varphi \in X$. Finally, let $X \vDash \varphi$ (read *from* X *follows* φ, or φ *is a consequence of* X) if every model $\mathcal{M}$ of X also satisfies the formula φ, i.e., $\mathcal{M} \vDash X \Rightarrow \mathcal{M} \vDash \varphi$.

As in Chapter 1, $\vDash$ denotes both the satisfaction and the consequence relation. Here, as there, we write $\varphi_1, \dots, \varphi_n \vDash \varphi$ for $\{\varphi_1, \dots, \varphi_n\} \vDash \varphi$. Note that in addition, $\vDash$ denotes the validity relation in structures, which is illustrated by the following

Example 2. We show that $\mathcal{A} \vDash \forall x \exists y\, x \neq y$, where the domain of $\mathcal{A}$ contains at least two elements. Indeed, let $\mathcal{M} = (\mathcal{A}, w)$ and let $a \in A$ be given arbitrarily. Then there exists some $b \in A$ with $a \neq b$. Hence, $(\mathcal{M}^a_x)^b_y = \mathcal{M}^{a\,b}_{xy} \vDash x \neq y$, and so $\mathcal{M}^a_x \vDash \exists y\, x \neq y$. Since a was arbitrary, $\mathcal{M} \vDash \forall x \exists y\, x \neq y$. Clearly the actual values of w are irrelevant in this argument. Hence $(\mathcal{A}, w) \vDash \forall x \exists y\, x \neq y$ for all w, that is, $\mathcal{A} \vDash \forall x \exists y\, x \neq y$.

Here some care is needed. While $\mathcal{M} \vDash \varphi$ or $\mathcal{M} \vDash \neg\varphi$ for all formulas, $\mathcal{A} \vDash \varphi$ or $\mathcal{A} \vDash \neg\varphi$ (the law of the excluded middle for validity in structures) is in general correct only for sentences φ, as Theorem 3.1 will show. If $\mathcal{A}$ contains more than one element, then, for example, neither $\mathcal{A} \vDash x = y$ nor $\mathcal{A} \vDash x \neq y$. Indeed, $x = y$ is falsified by any w such that $x^w \neq y^w$, and $x \neq y$ by any w with $x^w = y^w$. This is one of the reasons why models were not simply identified with structures.

For $\varphi \in \mathcal{L}$ let φ^g be the sentence $\forall x_1 \cdots \forall x_m \varphi$, where $x_1, \dots, x_m$ is an enumeration of $\mathit{free}\,\varphi$ according to index size, say. φ^g is called the *generalized of* φ, also called its *universal closure*. For $\varphi \in \mathcal{L}^0$ clearly $\varphi^g = \varphi$. From the definitions immediately results

(1) $\mathcal{A} \vDash \varphi \;\Leftrightarrow\; \mathcal{A} \vDash \varphi^g$,

and more generally, $\mathcal{A} \vDash X \Leftrightarrow \mathcal{A} \vDash X^g$ $(:= \{\varphi^g \mid \varphi \in X\})$. (1) explains why φ and φ^g are often notionally identified, and the information that formally runs φ^g is often shortened to φ. It must always be clear from the context whether our eye is on validity in a structure, or on validity in a model with its fixed valuation. Only in the first case can a generalization (or globalization) of the free variables be thought of as carried out. However, independent of this discussion, $\vDash \varphi \Leftrightarrow \vDash \varphi^g$ always holds.

Even after just these incomplete considerations it is already clear that numerous properties of structures and whole systems of axioms can adequately be described by first-order formulas and sentences. Thus, for example, an axiom system for groups in $\circ, e, ^{-1}$, mentioned already in **2.1**, can be formulated as follows:

$$\forall x\forall y\forall z\; x\circ(y\circ z)=(x\circ y)\circ z; \quad \forall x\; x\circ e=x; \quad \forall x\; x\circ x^{-1}=e.$$

Precisely, the sentences that follow from these axioms form the *elementary group theory in* $\circ, e, ^{-1}$. It will be denoted by $T_G^{=}$. In the sense elaborated in Exercise 3 in **2.6** an equivalent formulation of the theory of groups in $\circ, e$, denoted by T_G, is obtained if the third $T_G^{=}$-axiom is replaced by $\forall x\exists y\, x\circ y=e$. Let us mention that $\forall x\, e\circ x=x$ and $\forall x\exists y\, y\circ x=e$ are provable in T_G and also in $T_G^{=}$.

An axiom system for ordered sets can also easily be provided, in that one formalizes the properties of being irreflexive, transitive, and connex. Here and elsewhere, $\forall x_1\cdots x_n\varphi$ stands for $\forall x_1\cdots\forall x_n\varphi$:

$$\forall x\, x \not< x; \ \forall xyz(x<y\wedge y<z\rightarrow x<z); \ \forall xy(x\neq y\rightarrow x<y\vee y<x).$$

In writing down these and other axioms the outer $\forall$-prefixes are very often omitted so as to save on writing, and we think implicitly of the generalization of variables as having been carried out. This kind of economical writing is employed also in the formulation of (1) above, which strictly speaking runs 'for all $\mathcal{A}, \varphi : \mathcal{A}\vDash\varphi \Leftrightarrow \mathcal{A}\vDash\varphi^g$'.

For sentences α of a given language it is intuitively clear that the values of the variables of w for the relation $(\mathcal{A}, w)\vDash\alpha$ are irrelevant. The precise proof is extracted from the following theorem for $V=\emptyset$. Thus, either $(\mathcal{A}, w)\vDash\alpha$ for all w and hence $\mathcal{A}\vDash\alpha$, or else $(\mathcal{A}, w)\vDash\alpha$ for no w, i.e., $(\mathcal{A}, w)\vDash\neg\alpha$ for all w, and hence $\mathcal{A}\vDash\neg\alpha$. Sentences therefore obey the already-cited tertium non datur.

Theorem 3.1 (Coincidence theorem). *Let $V\subseteq$ Var, free $\varphi\subseteq V$, and $\mathcal{M}, \mathcal{M}'$ be models on the same domain A such that $x^{\mathcal{M}}=x^{\mathcal{M}'}$ for all $x\in V$, and $\mathsf{s}^{\mathcal{M}}=\mathsf{s}^{\mathcal{M}'}$ for all extralogical symbols s occurring in φ. Then $\mathcal{M}\vDash\varphi \Leftrightarrow \mathcal{M}'\vDash\varphi$.*

Proof by induction on φ. Let $\varphi = r\vec{t}$ be prime, so that $\mathit{var}\,\vec{t} \subseteq V$. As was mentioned earlier, the value of a term t depends only on the meaning of the symbols occurring in t. But in view of the suppositions, these meanings are the same in $\mathcal{M}$ and $\mathcal{M}'$. Therefore, $\vec{t}^{\,\mathcal{M}} = \vec{t}^{\,\mathcal{M}'}$ (i.e., $t_i^{\mathcal{M}} = t_i^{\mathcal{M}'}$ for $i = 1, \dots, n$), and so $\mathcal{M} \vDash r\vec{t} \Leftrightarrow r^{\mathcal{M}}\vec{t}^{\,\mathcal{M}} \Leftrightarrow r^{\mathcal{M}'}\vec{t}^{\,\mathcal{M}'} \Leftrightarrow \mathcal{M}' \vDash r\vec{t}$. For equations $t_1 = t_2$ one reasons analogously. Further, the induction hypothesis for α, β yields $\mathcal{M} \vDash \alpha \wedge \beta \Leftrightarrow \mathcal{M} \vDash \alpha, \beta \Leftrightarrow \mathcal{M}' \vDash \alpha, \beta \Leftrightarrow \mathcal{M}' \vDash \alpha \wedge \beta$. In the same way one obtains $\mathcal{M} \vDash \neg\alpha \Leftrightarrow \mathcal{M}' \vDash \neg\alpha$. By the induction step on $\forall$ it becomes clear that the induction hypothesis needs to be skillfully formulated. It must be given with respect to any pair $\mathcal{M}, \mathcal{M}'$ of models and any subset V of Var.

Therefore let $a \in A$ and $\mathcal{M}_x^a \vDash \varphi$. Since for $V' := V \cup \{x\}$ certainly $\mathit{free}\,\varphi \subseteq V'$ and the models $\mathcal{M}_x^a$, $\mathcal{M}'^a_x$ coincide for all $y \in V'$ (although in general $x^{\mathcal{M}} \neq x^{\mathcal{M}'}$), by the induction hypothesis $\mathcal{M}_x^a \vDash \varphi \Leftrightarrow \mathcal{M}'^a_x \vDash \varphi$, for each $a \in A$. This clearly implies

$$\mathcal{M} \vDash \forall x\varphi \Leftrightarrow \mathcal{M}_x^a \vDash \varphi \text{ for all } a \Leftrightarrow \mathcal{M}'^a_x \vDash \varphi \text{ for all } a \Leftrightarrow \mathcal{M}' \vDash \forall x\varphi. \qquad \square$$

It follows from this theorem that an $\mathcal{L}$-model $\mathcal{M} = (\mathcal{A}, w)$ of φ for the case that $\varphi \in \mathcal{L} \subseteq \mathcal{L}'$ can be completely arbitrarily expanded to an $\mathcal{L}'$-model $\mathcal{M}' = (\mathcal{A}', w)$ of φ, i.e., arbitrarily fixing $\mathsf{s}^{\mathcal{M}'}$ for $\mathsf{s} \in L' \backslash L$ gives $\mathcal{M} \vDash \varphi \Leftrightarrow \mathcal{M}' \vDash \varphi$ by the above theorem with $V = \mathit{Var}$. This readily implies that the consequence relation $\vDash_{\mathcal{L}'}$ with respect to $\mathcal{L}'$ is a *conservative* extension of $\vDash_{\mathcal{L}}$ in that $X \vDash_{\mathcal{L}} \varphi \Leftrightarrow X \vDash_{\mathcal{L}'} \varphi$, for all sets $X \subseteq \mathcal{L}$ and all $\varphi \in \mathcal{L}$. Hence, there is no need here for using indices. In particular, the satisfiability or general validity of φ depends only on the symbols effectively occurring in φ.

Another application of Theorem 3.1 is the following fact, which justifies the already mentioned "omission of superfluous quantifiers."

(2) $\forall x\varphi \equiv \varphi \equiv \exists x\varphi$ whenever $x \notin \mathit{free}\,\varphi$.

Indeed, $x \notin \mathit{free}\,\varphi$ implies $\mathcal{M} \vDash \varphi \;\Leftrightarrow\; \mathcal{M}_x^a \vDash \varphi$ (here $a \in A$ is arbitrary) according to Theorem 3.1; choose $\mathcal{M}' = \mathcal{M}_x^a$ and $V = \mathit{free}\,\varphi$. Therefore,

$$\begin{aligned}\mathcal{M} \vDash \forall x\varphi \Leftrightarrow \mathcal{M}_x^a \vDash \varphi \text{ for all } a \Leftrightarrow \mathcal{M} \vDash \varphi&\\ \Leftrightarrow \mathcal{M}_x^a \vDash \varphi \text{ for some } a \Leftrightarrow \mathcal{M} \vDash \exists x\varphi.&\end{aligned}$$

Very important for the next theorem and elsewhere is

(3) If $\mathcal{A} \subseteq \mathcal{B}$, $\mathcal{M} = (\mathcal{A}, w)$, $\mathcal{M}' = (\mathcal{B}, w)$ and $w\colon \mathit{Var} \to A$ then $t^{\mathcal{M}} = t^{\mathcal{M}'}$.

This is clear for prime terms, and the induction hypothesis $t_i^{\mathcal{M}} = t_i^{\mathcal{M}'}$ for $i = 1, \dots, n$ together with $f^{\mathcal{M}} = f^{\mathcal{M}'}$ imply

$$(f\vec{t}\,)^{\mathcal{M}} = f^{\mathcal{M}}(t_1^{\mathcal{M}}, \dots, t_n^{\mathcal{M}}) = f^{\mathcal{M}'}(t_1^{\mathcal{M}'}, \dots, t_n^{\mathcal{M}'}) = (f\vec{t}\,)^{\mathcal{M}'}.$$

For $\mathcal{M} = (\mathcal{A}, w)$ and $x_i^w = a_i$ let $t^{\mathcal{A},\vec{a}}$, or more suggestively $t^{\mathcal{A}}(\vec{a})$ denote the value of $t = t(\vec{x})$. Then (3) can somewhat more simply be written as

(4) $\mathcal{A} \subseteq \mathcal{B}$ *and* $t = t(\vec{x})$ *imply* $t^{\mathcal{A}}(\vec{a}) = t^{\mathcal{B}}(\vec{a})$ for all $\vec{a} \in A^n$.

Thus, along with the basic functions, also the so-called *term functions* $\vec{a} \mapsto t^{\mathcal{A}}(\vec{a})$ are the restrictions to their counterparts in $\mathcal{B}$. Clearly, if $n = 0$ or t is variable-free, one may write $t^{\mathcal{A}}$ for $t^{\mathcal{A}}(\vec{a})$. Note that in these cases $t^{\mathcal{A}} = t^{\mathcal{B}}$ whenever $\mathcal{A} \subseteq \mathcal{B}$, according to (4).

By Theorem 3.1 the satisfaction of φ in $(\mathcal{A}, w)$ depends only on the values of the $x \in \mathit{free}\,\varphi$. Let $\varphi = \varphi(\vec{x})$[4] and $\vec{a} = (a_1, \dots, a_n) \in A^n$. Then the statement

$$(\mathcal{A}, w) \vDash \varphi \text{ for a valuation } w \text{ with } x_1^w = a_1, \dots, x_n^w = a_n$$

can more suggestively be expressed by writing

$$(\mathcal{A}, \vec{a}) \vDash \varphi \quad \text{or} \quad \mathcal{A} \vDash \varphi\,[a_1, \dots, a_n] \quad \text{or} \quad \mathcal{A} \vDash \varphi\,[\vec{a}]$$

without mentioning w as a global valuation. Such notation also makes sense if w is restricted to a valuation on $\{x_1, \dots, x_n\}$. One may accordingly extend the concept of a model and call a pair $(\mathcal{A}, \vec{a})$ a model for a formula $\varphi(\vec{x})$ whenever $(\mathcal{A}, \vec{a}) \vDash \varphi(\vec{x})$, in particular if $\varphi \in \mathcal{L}^n$. We return to this extended concept in **4.1**. Until then we use it only for $n = 0$. That is, besides $\mathcal{M} = (\mathcal{A}, w)$ also the structure $\mathcal{A}$ itself is occasionally called a model for a set $S \subseteq \mathcal{L}^0$ of sentences, provided $\mathcal{A} \vDash S$.

As above let $\varphi = \varphi(\vec{x})$. Then $\varphi^{\mathcal{A}} := \{\vec{a} \in A^n \mid \mathcal{A} \vDash \varphi\,[\vec{a}]\}$ is called *the predicate defined by the formula* φ *in the structure* $\mathcal{A}$. For instance, the $\leqslant$-predicate in $(\mathbb{N}, +)$ is defined by $\varphi(x, y) = \exists z\, z + x = y$, but also by several other formulas.

More generally, a predicate $P \subseteq A^n$ is termed (explicitly or elementarily or first-order) *definable in* $\mathcal{A}$ if there is some $\varphi = \varphi(\vec{x})$ with $P = \varphi^{\mathcal{A}}$, and φ is called a *defining formula* for P. Analogously, $f\colon A^n \to A$ is called definable in $\mathcal{A}$ if $\varphi^{\mathcal{A}} = \operatorname{graph} f$ for some $\varphi(\vec{x}, y)$. One often talks in this

[4] Since this equation is to mean $\mathit{free}\,\varphi \subseteq \{x_1, \dots, x_n\}$, $\vec{x}$ is not uniquely determined by φ. Hence, the phrase "Let $\varphi = \varphi(\vec{x}) \dots$" implicitly includes along with a given φ also a tuple $\vec{x}$ given in advance. The notation $\varphi = \varphi(\vec{x})$ does not even state that φ contains free variables at all.

case of *explicit* definability of f in $\mathcal{A}$, to distinguish it from other kinds of definability. Much information is gained from the knowledge of which sets, predicates, or functions are definable in a structure. For instance, the sets definable in $(\mathbb{N}, 0, 1, +)$ are the eventually periodic ones (periodic from some number on). Thus, $\cdot$ cannot explicitly be defined by $+, 0, 1$ because the set of square numbers is not eventually periodic.

$\mathcal{A} \subseteq \mathcal{B}$ and $\varphi = \varphi(\vec{x})$ do not imply $\varphi^{\mathcal{A}} = \varphi^{\mathcal{B}} \cap A^n$, in general. For instance, let $\mathcal{A} = (\mathbb{N}, +)$, $\mathcal{B} = (\mathbb{Z}, +)$, and $\varphi = \exists z\, z + x = y$. Then $\varphi^{\mathcal{A}} = \leqslant^{\mathcal{A}}$, while $\varphi^{\mathcal{B}}$ contains all pairs $(a, b) \in \mathbb{Z}^2$. As the next theorem will show, $\varphi^{\mathcal{A}} = \varphi^{\mathcal{B}} \cap A^n$ holds in general only for open formulas φ, and is even characteristic for $\mathcal{A} \subseteq \mathcal{B}$ provided $A \subseteq B$. Clearly, $A \subseteq B$ is much weaker a condition than $\mathcal{A} \subseteq \mathcal{B}$:

Theorem 3.2 (Substructure theorem). *For structures $\mathcal{A}, \mathcal{B}$ such that $A \subseteq B$ the following conditions are equivalent:*

(i) $\mathcal{A} \subseteq \mathcal{B}$,

(ii) $\mathcal{A} \vDash \varphi\,[\vec{a}] \Leftrightarrow \mathcal{B} \vDash \varphi\,[\vec{a}]$, *for all open* $\varphi = \varphi(\vec{x})$ *and all* $\vec{a} \in A^n$,

(iii) $\mathcal{A} \vDash \varphi\,[\vec{a}] \Leftrightarrow \mathcal{B} \vDash \varphi\,[\vec{a}]$, *for all prime formulas* $\varphi(\vec{x})$ *and* $\vec{a} \in A^n$.

Proof. (i)⇒(ii): It suffices to prove that $\mathcal{M} \vDash \varphi \Leftrightarrow \mathcal{M}' \vDash \varphi$, with $\mathcal{M} = (\mathcal{A}, w)$ and $\mathcal{M}' = (\mathcal{B}, w)$, where $w\colon \mathit{Var} \to A$. In view of (3) the claim is obvious for prime formulas, and the induction steps for $\wedge, \neg$ are carried out just as in Theorem 3.1. (ii)⇒(iii): Trivial. (iii)⇒(i): By (iii), $r^{\mathcal{A}}\vec{a} \Leftrightarrow \mathcal{A} \vDash r\vec{x}\,[\vec{a}] \Leftrightarrow \mathcal{B} \vDash r\vec{x}\,[\vec{a}] \Leftrightarrow r^{\mathcal{B}}\vec{a}$. Analogously,

$$f^{\mathcal{A}}\vec{a} = b \;\Leftrightarrow\; \mathcal{A} \vDash f\vec{x} = y\,[\vec{a}, b] \;\Leftrightarrow\; \mathcal{B} \vDash f\vec{x} = y\,[\vec{a}, b] \;\Leftrightarrow\; f^{\mathcal{B}}\vec{a} = b,$$

for all $\vec{a} \in A^n$, $b \in A$. These conclusions state precisely that $\mathcal{A} \subseteq \mathcal{B}$. ❑

Let α be of the form $\forall\vec{x}\beta$ with open β, where $\forall\vec{x}$ may also be the empty prefix. Then α is a *universal* or $\forall$-*formula* (spoken “A-formula”), and for $\alpha \in \mathcal{L}^0$ also a *universal* or $\forall$-*sentence*. A simple example is $\forall x \forall y\; x = y$, which holds in $\mathcal{A}$ iff A contains precisely one element. Dually, $\exists\vec{x}\beta$ with β open is termed an $\exists$-*formula*, and an $\exists$-*sentence* whenever $\exists\vec{x}\beta \in \mathcal{L}^0$. Examples are the “how-many sentences”

$$\exists_1 := \exists \boldsymbol{v}_0\, \boldsymbol{v}_0 = \boldsymbol{v}_0; \quad \exists_n := \exists \boldsymbol{v}_0 \cdots \exists \boldsymbol{v}_{n-1} \textstyle\bigwedge_{i<j<n} \boldsymbol{v}_i \neq \boldsymbol{v}_j \quad (n > 1).$$

$\exists_n$ states ‘there exist at least n elements’, $\neg\exists_{n+1}$ thus that ‘there exist at most n elements’, and $\exists_{=n} := \exists_n \wedge \neg\exists_{n+1}$ says ‘there exist exactly

n elements'. Since $\exists_1$ is a tautology, it is convenient to set $\top := \exists_1$, and $\exists_0 := \bot := \neg\top$ in all first-order languages with equality. Clearly, equivalent definitions of $\top$, $\bot$ may be used as well.

Corollary 3.3. *Let $\mathcal{A} \subseteq \mathcal{B}$. Then every $\forall$-sentence $\forall\vec{x}\alpha$ valid in $\mathcal{B}$ is also satisfied in $\mathcal{A}$. Dually, every $\exists$-sentence $\exists\vec{x}\beta$ valid in $\mathcal{A}$ is also valid in $\mathcal{B}$.*

Proof. Let $\mathcal{B} \vDash \forall\vec{x}\beta$ and $\vec{a} \in A^n$. Then $\mathcal{B} \vDash \beta\,[\vec{a}]$, hence $\mathcal{A} \vDash \beta\,[\vec{a}]$ by Theorem 3.2. $\vec{a}$ was arbitrary and therefore $\mathcal{A} \vDash \forall\vec{x}\beta$. Now let $\mathcal{A} \vDash \exists\vec{x}\beta$. Then $\mathcal{A} \vDash \beta\,[\vec{a}]$ for some $\vec{a} \in A^n$, hence $\mathcal{B} \vDash \beta\,[\vec{a}]$ by Theorem 3.2, and consequently $\mathcal{B} \vDash \exists\vec{x}\beta$. ❑

We now formulate a generalization of certain individual often-used arguments about the invariance of properties under isomorphisms:

Theorem 3.4 (Invariance theorem). *Let $\mathcal{A}, \mathcal{B}$ be isomorphic structures of signature L and let $\imath\colon \mathcal{A} \to \mathcal{B}$ be an isomorphism. Then for all $\varphi = \varphi(\vec{x})$*

$$\mathcal{A} \vDash \varphi\,[\vec{a}] \;\Leftrightarrow\; \mathcal{B} \vDash \varphi\,[\imath\vec{a}] \quad \big(\vec{a} \in A^n,\ \imath\vec{a} = (\imath a_1, \dots, \imath a_n)\big).$$

In particular $\mathcal{A} \vDash \varphi \Leftrightarrow \mathcal{B} \vDash \varphi$, for all sentences φ of $\mathcal{L}$.

Proof. It is convenient to reformulate the claim as

$$\mathcal{M} \vDash \varphi \;\Leftrightarrow\; \mathcal{M}' \vDash \varphi \qquad \big(\mathcal{M} = (\mathcal{A}, w),\ \mathcal{M}' = (\mathcal{B}, w'),\ w' : x \mapsto \imath x^w\big).$$

This is easily confirmed by induction on φ after first proving $\imath(t^{\mathcal{M}}) = t^{\mathcal{M}'}$ inductively on t. This proof clearly includes the case $\varphi \in \mathcal{L}^0$. ❑

Thus, for example, it is once and for all clear that the isomorphic image of a group is a group even if we know at first only that it is a groupoid. Simply let φ in the theorem run through all axioms of group theory. Another application: Let $\imath$ be an isomorphism of the group $\mathcal{A} = (A, \circ)$ onto the group $\mathcal{A}' = (A', \circ)$ and let e and e' denote their unit elements, not named in the signature. We claim that nonetheless $\imath e = e'$, using the fact that the unit element of a group is the only solution of $x \circ x = x$ (Example 2, page 83). Thus, since $\mathcal{A} \vDash e \circ e = e$, we get $\mathcal{A}' \vDash \imath e \circ \imath e = \imath e$ by Theorem 3.4, hence $\imath e = e'$. Theorem 3.4, incidentally, holds for formulas of higher order as well. For instance, the property of being a continuously ordered set (formalizable in a second-order language, see **3.8**) is likewise invariant under isomorphism.

$\mathcal{L}$-structures $\mathcal{A}, \mathcal{B}$ are termed *elementarily equivalent* if $\mathcal{A} \vDash \alpha \Leftrightarrow \mathcal{B} \vDash \alpha$, for all $\alpha \in \mathcal{L}^0$. One then writes $\mathcal{A} \equiv \mathcal{B}$. We consider this important notion in **3.3** and more closely in **5.1**. Theorem 3.4 states in particular that $\mathcal{A} \simeq \mathcal{B} \Rightarrow \mathcal{A} \equiv \mathcal{B}$. The question immediately arises whether the converse of this also holds. For infinite structures the answer is negative (see **3.3**), for finite structures affirmative; a finite structure of a finite signature can, up to isomorphism, even be described by a single sentence. For example, the 2-element group $(\{0,1\},+)$ is up to isomorphism well determined by the following sentence, which tells us precisely how $+$ operates:

$$\exists \boldsymbol{v}_0 \exists \boldsymbol{v}_1 [\boldsymbol{v}_0 \boldsymbol{\neq} \boldsymbol{v}_1 \wedge \forall x(x \boldsymbol{=} \boldsymbol{v}_0 \vee x \boldsymbol{=} \boldsymbol{v}_1) \\ \wedge\ \boldsymbol{v}_0 + \boldsymbol{v}_0 \boldsymbol{=} \boldsymbol{v}_1 + \boldsymbol{v}_1 \boldsymbol{=} \boldsymbol{v}_0 \wedge \boldsymbol{v}_0 + \boldsymbol{v}_1 \boldsymbol{=} \boldsymbol{v}_1 + \boldsymbol{v}_0 \boldsymbol{=} \boldsymbol{v}_1].$$

We now investigate the behavior of the satisfaction relation under substitution. The definition of $\varphi\,\frac{t}{x}$ in **2.2** pays no attention to *collision of variables*, which is taken to mean that some variables of the substitution term t fall into the scope of quantifiers after the substitution has been performed. In this case $\mathcal{M} \vDash \forall x\varphi$ does not necessarily imply $\mathcal{M} \vDash \varphi\,\frac{t}{x}$, although this might have been expected. In other words, $\forall x\varphi \vDash \varphi\,\frac{t}{x}$ is not unrestrictedly correct. For instance, if $\varphi = \exists y\, x \boldsymbol{\neq} y$ then certainly $\mathcal{M} \vDash \forall x\varphi$ $(= \forall x \exists y\, x \boldsymbol{\neq} y)$ whenever $\mathcal{M}$ has at least two elements, but $\mathcal{M} \vDash \varphi\,\frac{y}{x}$ $(= \exists y\, y \boldsymbol{\neq} y)$ is certainly false. Analogously $\varphi\,\frac{t}{x} \vDash \exists x\varphi$ is not correct, in general. For example, choose $\forall y\, x \boldsymbol{=} y$ for φ and y for t.

One could forcibly obtain $\forall x\varphi \vDash \varphi\,\frac{t}{x}$ without any limitation by renaming bound variables by a suitable modification of the inductive definition of $\varphi\,\frac{t}{x}$ in the quantifier step. However, such measures are rather unwieldy for the arithmetization of proof method in **6.2**. It is therefore preferable to put up with minor restrictions when we are formulating rules of deduction later. The restrictions we will use are somewhat stronger than they need to be but can be handled more easily; they look as follows:

Call $\varphi, \frac{t}{x}$ *collision-free* if $y \notin bnd\,\varphi$ for all $y \in var\,t$ distinct from x. We need not require $x \notin bnd\,\varphi$ because t is substituted only at free occurrences of x in φ, that is, x cannot fall after substitution within the scope of a prefix $\forall x$, even if $x \in var\,t$. For collision-free $\varphi, \frac{t}{x}$ we always get $\forall x\varphi \vDash \varphi\,\frac{t}{x}$ by Corollary 3.6 below.

If σ is a global substitution (see **2.2**) then φ, σ are termed *collision-free* if $\varphi, \frac{x^\sigma}{x}$ are collision-free for every $x \in Var$. If $\sigma = \frac{\vec{t}}{\vec{x}}$, this condition clearly need be checked only for the pairs $\varphi, \frac{x^\sigma}{x}$ with $x \in var\,\vec{x}$ and $x \in free\,\varphi$.

For $\mathcal{M} = (\mathcal{A}, w)$ put $\mathcal{M}^\sigma := (\mathcal{A}, w^\sigma)$ with $x^{w^\sigma} := (x^\sigma)^\mathcal{M}$ for $x \in \mathit{Var}$, so that $x^{\mathcal{M}^\sigma} = x^{\sigma\mathcal{M}}$ $(= (x^\sigma)^\mathcal{M})$. This equation reproduces itself to

(5) $t^{\mathcal{M}^\sigma} = t^{\sigma\mathcal{M}}$ for all terms t.

Indeed, $t^{\mathcal{M}^\sigma} = f^\mathcal{M}(t_1^{\mathcal{M}^\sigma}, \dots, t_n^{\mathcal{M}^\sigma}) = f^\mathcal{M}(t_1^{\sigma\mathcal{M}}, \dots, t_n^{\sigma\mathcal{M}}) = t^{\sigma\mathcal{M}}$ for $t = f\vec{t}$ in view of the induction hypothesis $t_i^{\mathcal{M}^\sigma} = t_i^{\sigma\mathcal{M}}$ $(i = 1, \dots, n)$. Notice that $\mathcal{M}^\sigma$ coincides with $\mathcal{M}_{\vec{x}}^{\vec{t}^\mathcal{M}}$ for the case $\sigma = \frac{\vec{t}}{\vec{x}}$.

Theorem 3.5 (Substitution theorem). *Let $\mathcal{M}$ be a model and σ a global substitution. Then holds for all φ such that φ, σ are collision-free,*

(6) $\mathcal{M} \vDash \varphi^\sigma \;\Leftrightarrow\; \mathcal{M}^\sigma \vDash \varphi$.

In particular, $\mathcal{M} \vDash \varphi \frac{\vec{t}}{\vec{x}} \;\Leftrightarrow\; \mathcal{M}_{\vec{x}}^{\vec{t}^\mathcal{M}} \vDash \varphi$, provided $\varphi, \frac{\vec{t}}{\vec{x}}$ are collision-free.

Proof by induction on φ. In view of (5), we obtain

$$\mathcal{M} \vDash (t_1 = t_2)^\sigma \;\Leftrightarrow\; t_1^{\sigma\mathcal{M}} = t_2^{\sigma\mathcal{M}} \;\Leftrightarrow\; t_1^{\mathcal{M}^\sigma} = t_2^{\mathcal{M}^\sigma} \;\Leftrightarrow\; \mathcal{M}^\sigma \vDash t_1 = t_2.$$

Prime formulas $r\vec{t}$ are treated analogously. The induction steps for $\wedge, \neg$ in the proof of (6) are harmless. Only the $\forall$-step is interesting. The reader should recall the definition of $(\forall x\alpha)^\sigma$ page 60 and realize that the induction hypothesis refers to an arbitrary global substitution τ.

$$\begin{aligned}
\mathcal{M} \vDash (\forall x\alpha)^\sigma &\Leftrightarrow \mathcal{M} \vDash \forall x\, \alpha^\tau && (x^\tau = x \text{ and } y^\tau = y^\sigma \text{ else})\\
&\Leftrightarrow \mathcal{M}_x^a \vDash \alpha^\tau \text{ for all } a && (\text{definition})\\
&\Leftrightarrow (\mathcal{M}_x^a)^\tau \vDash \alpha \text{ for all } a && (\text{induction hypothesis})\\
&\Leftrightarrow (\mathcal{M}^\sigma)_x^a \vDash \alpha \text{ for all } a && \big((\mathcal{M}_x^a)^\tau = (\mathcal{M}^\sigma)_x^a,\ \text{see below}\big)\\
&\Leftrightarrow \mathcal{M}^\sigma \vDash \forall x\alpha.
\end{aligned}$$

We show that $(\mathcal{M}_x^a)^\tau = (\mathcal{M}^\sigma)_x^a$. Since $\forall x\alpha, \sigma$ (hence $\forall x\alpha, \frac{y^\sigma}{y}$ for every y) are collision-free, we have $x \notin \mathit{var}\, y^\sigma$ if $y \neq x$, and since $y^\tau = y^\sigma$ we get in this case $y^{(\mathcal{M}_x^a)^\tau} = y^{\tau\mathcal{M}_x^a} = y^{\sigma\mathcal{M}_x^a} = y^{\sigma\mathcal{M}} = y^{\mathcal{M}^\sigma} = y^{(\mathcal{M}^\sigma)_x^a}$. But also in the case $y = x$ we have $x^{(\mathcal{M}_x^a)^\tau} = x^{\tau\mathcal{M}_x^a} = x^{\mathcal{M}_x^a} = a = x^{(\mathcal{M}^\sigma)_x^a}$. ❏

Corollary 3.6. *For all φ and $\frac{\vec{t}}{\vec{x}}$ such that $\varphi, \frac{\vec{t}}{\vec{x}}$ are collision-free, the following properties hold:*

(a) $\forall\vec{x}\varphi \vDash \varphi \frac{\vec{t}}{\vec{x}}$, *in particular* $\forall x\varphi \vDash \varphi \frac{t}{x}$, (b) $\varphi \frac{\vec{t}}{\vec{x}} \vDash \exists\vec{x}\varphi$,

(c) $\varphi \frac{s}{x}, s = t \vDash \varphi \frac{t}{x}$, *provided $\varphi, \frac{s}{x}, \frac{t}{x}$ are collision-free.*

Proof. Let $\mathcal{M} \vDash \forall\vec{x}\varphi$, so that $\mathcal{M}_{\vec{x}}^{\vec{a}} \vDash \varphi$ for all $\vec{a} \in A^n$. In particular, $\mathcal{M}_{\vec{x}}^{\vec{t}^\mathcal{M}} \vDash \varphi$. Therefore, $\mathcal{M} \vDash \varphi \frac{\vec{t}}{\vec{x}}$ by Theorem 3.5. (b) follows easily from $\neg\exists\vec{x}\varphi \vDash \neg\varphi \frac{\vec{t}}{\vec{x}}$. This holds by (a), for $\neg\exists\vec{x}\varphi \equiv \forall\vec{x}\neg\varphi$ and $\neg(\varphi \frac{\vec{t}}{\vec{x}}) \equiv (\neg\varphi) \frac{\vec{t}}{\vec{x}}$.

(c): Let $\mathcal{M} \vDash \varphi\,\frac{s}{x}, s = t$, so that $s^{\mathcal{M}} = t^{\mathcal{M}}$ and $\mathcal{M}_x^{s^{\mathcal{M}}} \vDash \varphi$ by the theorem. Clearly, then also $\mathcal{M}_x^{t^{\mathcal{M}}} \vDash \varphi$. Hence $\mathcal{M} \vDash \varphi\,\frac{t}{x}$. ❑

Remark 2. The identical substitution ι is obviously collision-free with every formula. Thus, $\forall x\varphi \vDash \varphi \ (= \varphi^{\iota})$ is always the case, while $\forall x\varphi \vDash \varphi\,\frac{t}{x}$ is correct in general only if t contains at most the variable x, since $\varphi, \frac{t}{x}$ are then collision-free. Theorem 3.5 and Corollary 3.6 are easily strengthened. Define inductively a ternary predicate 't is free for x in φ', which intuitively is to mean that no free occurrence in φ of the variable x lies within the scope of a prefix $\forall y$ whenever $y \in \mathit{var}\,t$. In this case Theorem 3.5 holds for $\sigma = \frac{t}{x}$ as well, so that nothing needs to be changed in the proofs based on this theorem if one works with 't is free for x in φ', or simply reads "$\varphi, \frac{t}{x}$ are collision-free" as "t is free for x in φ." Though collision-freeness is somewhat cruder and slightly more restrictive, it is for all that more easily manageable, which will pay off, for example, in **6.2**, where proofs will be arithmetized. Once one has become accustomed to the required caution, it is allowable not always to state explicitly the restrictions caused by collisions of variables, but rather to assume them tacitly.

Theorem 3.5 also shows that the quantifier "there exists exactly one," denoted by $\exists!$, is correctly defined by $\exists! x\varphi := \exists x\varphi \wedge \forall x\forall y(\varphi \wedge \varphi\,\frac{y}{x} \to x = y)$ with $y \notin \mathit{var}\,\varphi$. Indeed, it is easily seen that $\mathcal{M} \vDash \forall x\forall y(\varphi \wedge \varphi\,\frac{y}{x} \to x = y)$ means just $\mathcal{M}_x^a \vDash \varphi \ \&\ \mathcal{M}_y^b \vDash \varphi\,\frac{y}{x} \ \Rightarrow\ a = b$. In short, $\mathcal{M}_x^a \vDash \varphi$ for at most one a. Putting everything together, $\mathcal{M} \vDash \exists! x\varphi$ iff there is precisely one $a \in A$ with $\mathcal{M}_x^a \vDash \varphi$. An example is $\mathcal{M} \vDash \exists! x\, x = t$ for arbitrary $\mathcal{M}$ and $x \notin \mathit{var}\,t$. In other words, $\exists! x\, x = t$ is a tautology. Half of this, namely $\vDash \exists x\, x = t$, was shown in Example 1, and $\vDash \forall x\forall y(x = t \wedge y = t \to x = y)$ is obvious. There are various equivalent definitions of $\exists! x\varphi$. For example, a short and catchy formula is $\exists x\forall y(\varphi\,\frac{y}{x} \leftrightarrow x = y)$, where $y \notin \mathit{var}\,\varphi$. The equivalence proof is left to the reader.

Exercises

1. Let $X \vDash \varphi$ and $x \notin \mathit{free}\,X$. Show that $X \vDash \forall x\varphi$.

2. Prove that $\forall x(\alpha \to \beta) \vDash \forall x\alpha \to \forall x\beta$, which is obviously equivalent to $\vDash \forall x(\alpha \to \beta) \to \forall x\alpha \to \forall x\beta$.

3. Suppose $\mathcal{A}'$ results from $\mathcal{A}$ by adjoining a constant symbol $\boldsymbol{a}$ for some $a \in A$. Prove $\mathcal{A} \vDash \alpha\,[a] \Leftrightarrow \mathcal{A}' \vDash \alpha(\boldsymbol{a})\ (= \alpha\,\frac{\boldsymbol{a}}{x})$ for $\alpha = \alpha(x)$, by first verifying $t(x)^{\mathcal{A},a} = t(\boldsymbol{a})^{\mathcal{A}'}$. This is easily generalized to the case of more than one free variable in α.

4. Show that (a) A conjunction of the $\exists_i$ and their negations is equivalent to $\exists_n \wedge \neg\exists_m$ for suitable n, m ($\exists_n \wedge \neg\exists_0 \equiv \exists_n$, $\exists_1 \wedge \neg\exists_m \equiv \neg\exists_m$). (b) A Boolean combination of the $\exists_i$ is equivalent to $\bigvee_{\nu\leqslant n} \exists_{=k_\nu}$ or to $\exists_k \vee \bigvee_{\nu\leqslant n} \exists_{=k_\nu}$, with $k_0 < \cdots < k_n < k$. Note that $\bigvee_{\nu\leqslant n} \exists_{=k_\nu}$ equals $\exists_{=0}$ ($\equiv \bot$) for $n{=}k_0{=}0$ and $\neg\exists_n \equiv \bigvee_{\nu<n} \exists_{=\nu}$ for $n{>}0$.

2.4 General Validity and Logical Equivalence

From the perspective of predicate logic $\alpha \vee \neg\alpha$ ($\alpha \in \mathcal{L}$) is a trivial example of a tautology, because it results by inserting α for p from the propositional tautology $p \vee \neg p$. Every propositional tautology provides generally valid $\mathcal{L}$-formulas by the insertion of $\mathcal{L}$-formulas for the propositional variables. But there are tautologies not arising in this way. $\forall x(x < x \vee x \nless x)$ is an example, though it has still a root in propositional logic. Tautologies without a such a root are $\exists x\, x = x$ and $\exists x\, x = t$ for $x \notin \mathit{var}\, t$. The former arises from the convention that structures are always nonempty, the latter from the restriction to totally defined basic operations. A particularly interesting tautology is given by the following

Example 1 (Russell's antinomy). We will show that the "Russellian set" u, consisting of all sets not containing themselves as a member, does not exist which clearly follows from $\vDash \neg\exists u\forall x(x \in u \leftrightarrow x \notin x)$. We start with $\forall x(x \in u \leftrightarrow x \notin x) \vDash u \in u \leftrightarrow u \notin u$. This holds by Corollary 3.6(a). Clearly, $u \in u \leftrightarrow u \notin u$ is unsatisfiable. Hence, the same holds for $\forall x(x \in u \leftrightarrow x \notin x)$, and thus for $\exists u\forall x(x \in u \leftrightarrow x \notin x)$. Consequently, $\vDash \neg\exists u\forall x(x \in u \leftrightarrow x \notin x)$.

Note that we need not assume in the above argument that $\in$ means membership. The proof of $\vDash \neg\exists u\forall x(x \in u \leftrightarrow x \notin x)$ need not be related to set theory at all. Hence, our example represents rather a logical paradox than a set-theoretic antinomy. What looks like an antinomy here is the expectation that $\exists u\forall x(x \in u \leftrightarrow x \notin x)$ *should* hold in set theory if $\in$ is to mean membership and Cantor's definition of a set is taken literally.

The satisfaction clause for $\alpha \to \beta$ easily yields $\alpha \vDash \beta \ \Leftrightarrow\ \vDash \alpha \to \beta$, a special case of $X, \alpha \vDash \beta \ \Leftrightarrow\ X \vDash \alpha \to \beta$. This can be very useful in checking whether formulas given in implicative form are tautologies, as was mentioned already in **1.3**. For instance, from $\forall x\alpha \vDash \alpha\,\frac{t}{x}$ (which holds for collision-free $\alpha, \frac{t}{x}$) we immediately get $\vDash \forall x\alpha \to \alpha\,\frac{t}{x}$.

As in propositional logic, $\alpha \equiv \beta$ is again equivalent to $\vDash \alpha \leftrightarrow \beta$. By inserting $\mathcal{L}$-formulas for the variables of a propositional equivalence one automatically procures one of predicate logic. Thus, for instance, $\alpha \to \beta \equiv \neg\alpha \vee \beta$, because certainly $p \to q \equiv \neg p \vee q$. Since every $\mathcal{L}$-formula results from the insertion of propositionally irreducible $\mathcal{L}$-formulas in a formula of propositional logic, one also sees that every $\mathcal{L}$-formula can be converted into a conjunctive normal form. But there are also numerous other equivalences, for example $\neg\forall x\alpha \equiv \exists x\neg\alpha$ and $\neg\exists x\alpha \equiv \forall x\neg\alpha$. The first of these means just $\neg\forall x\alpha \equiv \neg\forall x\neg\neg\alpha$ $(= \exists x\neg\alpha)$, obtained by replacing α by the equivalent formula $\neg\neg\alpha$ under the prefix $\forall x$. This is a simple application of Theorem 4.1 below with $\equiv$ for $\approx$.

As in propositional logic, semantic equivalence is an equivalence relation in $\mathcal{L}$ and, moreover, a *congruence in* $\mathcal{L}$. Speaking more generally, an equivalence relation $\approx$ in $\mathcal{L}$ satisfying the congruence property

$$\text{CP:}\quad \alpha \approx \alpha',\ \beta \approx \beta' \ \Rightarrow\ \alpha \wedge \beta \approx \alpha' \wedge \beta',\ \neg\alpha \approx \neg\alpha',\ \forall x\alpha \approx \forall x\alpha'$$

is termed a *congruence in* $\mathcal{L}$. Its most important property is expressed by

Theorem 4.1 (Replacement theorem). *Let $\approx$ be a congruence in $\mathcal{L}$ and $\alpha \approx \alpha'$. If φ' results from φ by replacing the formula α at one or more of its occurrences in φ by the formula α', then $\varphi \approx \varphi'$.*

Proof by induction on φ. Suppose φ is a prime formula. Both for $\varphi = \alpha$ and $\varphi \neq \alpha$, $\varphi \approx \varphi'$ clearly holds. Now let $\varphi = \varphi_1 \wedge \varphi_2$. In case $\varphi = \alpha$ holds trivially $\varphi \approx \varphi'$. Otherwise $\varphi' = \varphi_1' \wedge \varphi_2'$, where φ_1', φ_2' result from φ_1, φ_1 by possible replacements. By the induction hypothesis $\varphi_1 \approx \varphi_1'$ and $\varphi_2 \approx \varphi_2'$. Hence, $\varphi = \varphi_1 \wedge \varphi_2 \approx \varphi_1' \wedge \varphi_2' = \varphi'$ according to CP above. The induction steps for $\neg$, $\forall$ follow analogously. ❏

This theorem will constantly be used, mainly with $\equiv$ for $\approx$, without actually specifically being cited, just as in the arithmetical rearrangement of terms, where the laws of arithmetic used are hardly ever named explicitly. The theorem readily implies that CP is provable for all defined connectives such as $\to$ and $\exists$. For example, $\alpha \approx \alpha' \Rightarrow \exists x\alpha \approx \exists x\alpha'$, because $\alpha \approx \alpha' \Rightarrow \exists x\alpha = \neg\forall x\neg\alpha \approx \neg\forall x\neg\alpha' = \exists x\alpha'$.

First-order languages have a finer structure than those of propositional logic. There are consequently further interesting congruences in $\mathcal{L}$. In particular, formulas α, β are *equivalent in an $\mathcal{L}$-structure $\mathcal{A}$*, in symbols

$\alpha \equiv_{\mathcal{A}} \beta$, if $\mathcal{A} \vDash \alpha\,[w] \Leftrightarrow \mathcal{A} \vDash \beta\,[w]$, for all w. Hence, in $\mathcal{A} = (\mathbb{N}, <, +, 0)$ the formulas $x < y$ and $\exists z\,(z \neq 0 \wedge x + z = y)$ are equivalent. The proof of CP for $\equiv_{\mathcal{A}}$ is very simple and is therefore left to the reader.

Clearly, $\alpha \equiv_{\mathcal{A}} \beta$ is equivalent to $\mathcal{A} \vDash \alpha \leftrightarrow \beta$. Because of $\equiv\; \subseteq\; \equiv_{\mathcal{A}}$, properties such as $\neg\forall x\alpha \equiv \exists x\neg\alpha$ carry over from $\equiv$ to $\equiv_{\mathcal{A}}$. But there are often new interesting equivalences in certain structures. For instance, there are structures in which every formula is equivalent to a formula without quantifiers, as we will see in **5.6**.

A very important fact with an almost trivial proof is that the intersection of a family of congruences is itself a congruence. Consequently, for any class $\boldsymbol{K} \neq \emptyset$ of $\mathcal{L}$-structures, $\equiv_{\boldsymbol{K}} := \bigcap\{\equiv_{\mathcal{A}} \mid \mathcal{A} \in \boldsymbol{K}\}$ is necessarily a congruence. For the class $\boldsymbol{K}$ of *all* $\mathcal{L}$-structures, $\equiv_{\boldsymbol{K}}$ equals the logical equivalence $\equiv$, which in this section we deal with exclusively. Below we list its most important features; these should be committed to memory, since they will continually be applied.

(1) $\forall x(\alpha \wedge \beta) \equiv \forall x\alpha \wedge \forall x\beta$, (2) $\exists x(\alpha \vee \beta) \equiv \exists x\alpha \vee \exists x\beta$,

(3) $\forall x\forall y\alpha \equiv \forall y\forall x\alpha$, (4) $\exists x\exists y\alpha \equiv \exists y\exists x\alpha$.

If x does not occur free in the formula β, then also

(5) $\forall x(\alpha \vee \beta) \equiv \forall x\alpha \vee \beta$, (6) $\exists x(\alpha \wedge \beta) \equiv \exists x\alpha \wedge \beta$,

(7) $\forall x\beta \equiv \beta$, (8) $\exists x\beta \equiv \beta$,

(9) $\forall x(\alpha \rightarrow \beta) \equiv \exists x\alpha \rightarrow \beta$, (10) $\exists x(\alpha \rightarrow \beta) \equiv \forall x\alpha \rightarrow \beta$.

The simple proofs are left to the reader. (7) and (8) were stated in (2) in **2.3**. Only (9) and (10) look at first sight surprising. But in practice these equivalences are very frequently used. For instance, consider for a fixed set of formulas X the evidently true metalogical assertion 'for all α: if $X \vDash \alpha, \neg\alpha$ then $X \vDash \forall x\, x \neq x$'. This clearly states the same as 'If there is some α such that $X \vDash \alpha, \neg\alpha$ then $X \vDash \forall x\, x \neq x$'.

Remark. In everyday speech variables tend to remain unquantified, partly because in some cases the same meaning results from quantifying with "there exists a" as with "for all." For instance, consider the following three sentences, which obviously tell us the same thing, and of which the last two correspond to the logical equivalence (9):

- If a lawyer finds a loophole in the law it must be changed.
- If there is a lawyer who finds a loophole in the law it must be changed.
- For all lawyers: if one of them finds a loophole in the law then it must be changed.

Often, the type of quantification in linguistic bits of information can be made out only from the context, and this leads not all too seldom to unintentional (or intentional) misunderstandings. “Logical relations in language are almost always just alluded to, left to guesswork, and not actually expressed” (G. Frege).

Let x, y be distinct variables and $\alpha \in \mathcal{L}$. One of the most important logical equivalences is *renaming of bound variables* (in short, *bound renaming*), stated in

(11) (a) $\forall x\alpha \equiv \forall y(\alpha\,\frac{y}{x})$, (b) $\exists x\alpha \equiv \exists y(\alpha\,\frac{y}{x})$ $(y \notin \mathit{var}\,\alpha)$.

(b) follows from (a) by rearranging equivalently. Note that $y \notin \mathit{var}\,\alpha$ is equivalent to $y \notin \mathit{free}\,\alpha$ and $\alpha, \frac{y}{x}$ collision-free. Writing $\mathcal{M}^{y}_{x}$ for $\mathcal{M}^{y^{\mathcal{M}}}_{x}$, (a) derives as follows:

$$\begin{array}{rlll}
\mathcal{M} \vDash \forall x\alpha & \Leftrightarrow\ \mathcal{M}^{a}_{x} \vDash \alpha & \text{for all } a & (\text{definition})\\
& \Leftrightarrow\ (\mathcal{M}^{a}_{y})^{a}_{x} \vDash \alpha & \text{for all } a & (\text{Theorem } 3.1)\\
& \Leftrightarrow\ (\mathcal{M}^{a}_{y})^{y}_{x} \vDash \alpha & \text{for all } a & \big((\mathcal{M}^{a}_{y})^{y}_{x} = (\mathcal{M}^{a}_{y})^{a}_{x}\big)\\
& \Leftrightarrow\ \mathcal{M}^{a}_{y} \vDash \alpha\,\frac{y}{x} & \text{for all } a & (\text{Theorem } 3.5)\\
& \Leftrightarrow\ \mathcal{M} \vDash \forall y(\alpha\frac{y}{x})\,. & &
\end{array}$$

(12) and (13) below are also noteworthy. According to (13), substitutions are completely described up to logical equivalence by so-called *free renamings* (substitutions of the form $\frac{y}{x}$). (13) also embraces the case $x \in \mathit{var}\,t$. In (12) and (13) we tacitly assume that $\alpha, \frac{t}{x}$ are collision-free.

(12) $\forall x(x = t \to \alpha) \equiv \alpha\,\frac{t}{x} \equiv \exists x(x = t \land \alpha)$ $(x \notin \mathit{var}\,t)$.

(13) $\forall y(y = t \to \alpha\,\frac{y}{x}) \equiv \alpha\,\frac{t}{x} \equiv \exists y(y = t \land \alpha\,\frac{y}{x})$ $(y \notin \mathit{var}\,\alpha, t)$.

Proof of (12): $\forall x(x = t \to \alpha) \vDash (x = t \to \alpha)\,\frac{t}{x} = t = t \to \alpha\,\frac{t}{x} \vDash \alpha\,\frac{t}{x}$ by Corollary 3.6. Conversely, let $\mathcal{M} \vDash \alpha\,\frac{t}{x}$. If $\mathcal{M}^{a}_{x} \vDash x = t$ then clearly $a = t^{\mathcal{M}}$. Hence also $\mathcal{M}^{a}_{x} \vDash \alpha$, since $\mathcal{M}^{t^{\mathcal{M}}}_{x} \vDash \alpha$. Thus, $\mathcal{M}^{a}_{x} \vDash x = t \to \alpha$ for any $a \in A$, i.e., $\mathcal{M} \vDash \forall x(x = t \to \alpha)$. This proves the left equivalence in (12). The right equivalence reduces to the left one because

$$\exists x(x = t \land \alpha) = \neg\forall x\neg(x = t \land \alpha) \equiv \neg\forall x(x = t \to \neg\alpha) \equiv \neg\neg\alpha\,\frac{t}{x} \equiv \alpha\,\frac{t}{x}\ .$$

Item (13) is proved similarly. Note that $\forall y(y = t \to \alpha\,\frac{y}{x}) \vDash \alpha\,\frac{y}{x}\,\frac{t}{y} = \alpha\,\frac{t}{x}$ by Corollary 3.6 and Exercise 4 in **2.2**.

With the above equivalences we can now regain an equivalent formula starting with any formula in which all quantifiers are standing at the beginning. But this result requires both quantifiers $\forall$ and $\exists$, in the following denoted by $\mathtt{Q}, \mathtt{Q}_1, \mathtt{Q}_2, \ldots$

A formula of the form $\alpha = \mathtt{Q}_1 x_1 \cdots \mathtt{Q}_n x_n \beta$ with an open formula β is termed a *prenex formula* or a *prenex normal form*, in short, a PNF. β is called the *kernel* of α. W.l.o.g. $x_1, \dots, x_n$ are distinct and x_i occurs free in β since we may drop "superfluous quantifiers," see (2) page 66. Prenex normal forms are very important for classifying definable number-theoretic predicates in **6.3**, and for other purposes. The already mentioned $\forall$- and $\exists$-formulas are the simplest examples.

Theorem 4.2 (on the prenex normal form). *Every formula φ is equivalent to a formula in prenex normal form that can effectively be constructed from φ.*

Proof. Without loss of generality let φ contain (besides $=$) at most the logical symbols $\neg, \wedge, \forall, \exists$. For each prefix $\mathtt{Q}x$ in φ consider the number of symbols $\neg$ or left parentheses occurring to the left of $\mathtt{Q}x$. Let $s\varphi$ be the sum of these numbers, summed over all prefixes occurring in φ. Clearly, φ is a PNF iff $s\varphi = 0$. Let $s\varphi \neq 0$. Then φ contains some prefix $\mathtt{Q}x$ and $\neg$ or $\wedge$ stands immediately in front of $\mathtt{Q}x$. A successive application of either $\neg\forall x\alpha \equiv \exists x\neg\alpha$, $\neg\exists x\alpha \equiv \forall x\neg\alpha$, or $(\beta \wedge \mathtt{Q}x\alpha) \equiv \mathtt{Q}y(\beta \wedge \alpha\frac{y}{x})$ $(y \notin \mathit{var}\,\alpha, \beta)$, inside φ obviously reduces $s\varphi$ stepwise. ❑

Example 2. $\forall x\exists y(x \neq 0 \to x \cdot y = 1)$ is a PNF for $\forall x(x \neq 0 \to \exists y\, x \cdot y = 1)$. And $\exists x\forall y\forall z(\varphi \wedge (\varphi\frac{y}{x} \wedge \varphi\frac{z}{x} \to y = z))$ for $\exists x\varphi \wedge \forall y\forall z(\varphi\frac{y}{x} \wedge \varphi\frac{z}{x} \to y = z)$, provided $y, z \notin \mathit{free}\,\varphi$; if not, a bound renaming will help. An equivalent PNF for this formula with minimal quantifier rank is $\exists x\forall y(\varphi\frac{y}{x} \leftrightarrow x = y)$.

The formula $\forall x(x \neq 0 \to \exists y\, x \cdot y = 1)$ from Example 2 may be abbreviated by $(\forall x \neq 0)\exists y\, x \cdot y = 1$. More generally, we shall often write $(\forall x \neq t)\alpha$ for $\forall x(x \neq t \to \alpha)$ and $(\exists x \neq t)\alpha$ for $\exists x(x \neq t \wedge \alpha)$. A similar notation is used for $\leqslant$, $<$, $\in$ and their negations. For instance, $(\forall x \leqslant t)\alpha$ and $(\exists x \leqslant t)\alpha$ are to mean $\forall x(x \leqslant t \to \alpha)$ and $\exists x(x \leqslant t \wedge \alpha)$, respectively. For any binary relation symbol $\lhd$, the "prefixes" $(\forall y \lhd x)$ and $(\exists y \lhd x)$ are related to each other, as are $\forall$ and $\exists$, see Exercise 2.

Exercises

1. Let $\alpha \equiv \beta$. Prove that $\alpha\frac{\vec{t}}{\vec{x}} \equiv \beta\frac{\vec{t}}{\vec{x}}$ $\quad(\alpha, \frac{\vec{t}}{\vec{x}}$ and $\beta, \frac{\vec{t}}{\vec{x}}$ collision-free).

2. Prove that $\neg(\forall x \lhd y)\alpha \equiv (\exists x \lhd y)\neg\alpha$ and $\neg(\exists x \lhd y)\alpha \equiv (\forall x \lhd y)\neg\alpha$. Here $\lhd$ represents any binary relation symbol.

3. Show by means of bound renaming that both the conjunction and the disjunction of $\forall$-formulas α, β is equivalent to some $\forall$-formula. Prove the same for $\exists$-formulas.

4. Show that every formula $\varphi \in \mathcal{L}$ is equivalent to some $\varphi' \in \mathcal{L}$ built up from literals by means of $\wedge$, $\vee$, $\forall$, and $\exists$.

5. Let P be a unary predicate symbol. Prove that $\exists x(Px \rightarrow \forall yPy)$ is a tautology.

6. Call $\alpha, \beta \in \mathcal{L}$ *tautologically equivalent* if $\vDash \alpha \Leftrightarrow \vDash \beta$. Confirm that the following (in general not logically equivalent) formulas are tautologically equivalent: α, $\forall x\alpha$, and $\alpha\,\frac{c}{x}$, where the constant symbol c does not occur in α.

2.5 Logical Consequence and Theories

Whenever $\mathcal{L}' \supseteq \mathcal{L}$, the language $\mathcal{L}'$ is called an *expansion* or *extension* of $\mathcal{L}$ and $\mathcal{L}$ a *reduct* or *restriction* of $\mathcal{L}'$. Recall the insensitivity of the consequence relation to extensions of a first-order language, mentioned in **2.3**. Theorem 3.1 yields that establishing $X \vDash \alpha$ does not depend on the language to which the set of formulas X and the formula α belong. For this reason, indices for $\vDash$, such as $\vDash_{\mathcal{L}}$, are dispensable.

Because of the unaltered satisfaction conditions for $\wedge$ and $\neg$, all properties of the propositional consequence gained in **1.3** carry over to the first-order logical consequence relation. These include general properties such as, for example, the reflexivity and transitivity of $\vDash$, and the semantic counterparts of the rules $(\wedge 1)$, $(\wedge 2)$, $(\neg 1)$, $(\neg 2)$ from **1.4**, for instance the counterpart of $(\wedge 1)$, $\dfrac{X \vDash \alpha, \beta}{X \vDash \alpha \wedge \beta}$.[5]

In addition, Gentzen-style properties such as the deduction theorem automatically carry over. But there are also completely new properties. Some of these will be elevated to basic rules of a logical calculus for first-order languages in **3.1**, to be found among the following ones:

[5] A suggestive way of writing "$X \vDash \alpha, \beta$ implies $X \vDash \alpha \wedge \beta$," a notation that was introduced already in Exercise 3 in **1.3**. A corresponding notation will also be used in stating the properties of $\vDash$ on the next page.

Some properties of the predicate logical consequence relation.

(a) $\dfrac{X \vDash \forall x\alpha}{X \vDash \alpha\,\frac{t}{x}}$ $\quad(\alpha, \frac{t}{x}$ collision-free),

(b) $\dfrac{X \vDash \alpha\,\frac{s}{x},\, s=t}{X \vDash \alpha\,\frac{t}{x}}$ $\quad(\alpha, \frac{s}{x}$ and $\alpha, \frac{t}{x}$ collision-free),

(c) $\dfrac{X, \beta \vDash \alpha}{X, \forall x\beta \vDash \alpha}$ $\quad$(anterior generalization),

(d) $\dfrac{X \vDash \alpha}{X \vDash \forall x\alpha}$ $\quad(x \notin \mathit{free}\,X$, posterior generalization),

(e) $\dfrac{X, \beta \vDash \alpha}{X, \exists x\beta \vDash \alpha}$ $\quad(x \notin \mathit{free}\,X, \alpha$, anterior particularization),

(f) $\dfrac{X \vDash \alpha\,\frac{t}{x}}{X \vDash \exists x\alpha}$ $\quad(\alpha, \frac{t}{x}$ collision-free, posterior particularization)

(a) follows from $X \vDash \forall x\alpha \vDash \alpha\,\frac{t}{x}$, for $\vDash$ is transitive. Similarly, (b) follows from $\alpha\,\frac{s}{x}, s=t \vDash \alpha\,\frac{t}{x}$, stated in Corollary 3.6. Analogously (c) results from $\forall x\beta \vDash \beta$. To prove (d), suppose that $X \vDash \alpha$, $\mathcal{M} \vDash X$, and $x \notin \mathit{free}\,X$. Then $\mathcal{M}^a_x \vDash X$ for any $a \in A$ by Theorem 3.1, which just means $\mathcal{M} \vDash \forall x\alpha$. As regards (e), let $X, \beta \vDash \alpha$. Observe that by contraposition and by (d),

$$X, \beta \vDash \alpha \;\Rightarrow\; X, \neg\alpha \vDash \neg\beta \;\Rightarrow\; X, \neg\alpha \vDash \forall x\neg\beta,$$

whence $X, \neg\forall x\neg\beta \vDash \alpha$. (e) captures deduction *from* an existence claim, while (f) *confirms* an existence claim. (f) holds since $\alpha\,\frac{t}{x} \vDash \exists x\alpha$ according to Corollary 3.6. Both (e) and (f) are permanently applied in mathematical reasoning and will briefly be discussed in Example 1 on the next page. All above properties have certain variants; for example, a variant of (d) is

(g) $\dfrac{X \vDash \alpha\,\frac{y}{x}}{X \vDash \forall x\alpha}$ $\quad(y \notin \mathit{free}\,X \cup \mathit{var}\,\alpha)$.

This results from (d) with $\alpha\,\frac{y}{x}$ for α and y for x, since $\forall y\alpha\,\frac{y}{x} \equiv \forall x\alpha$.

From the above properties, complicated chains of deduction can, where necessary, be justified step by step. But in practice this makes sense only in particular circumstances, because formalized proofs are readable only at the expense of a lot of time, just as with lengthy computer programs, even with well-prepared documentation. What is most important is that a proof, when written down, can be understood and reproduced. This is why mathematical deduction tends to proceed *informally*, i.e., both claims and their proofs are formulated in a mathematical "everyday" language with the aid of fragmentary and flexible formalization. To what degree a proof is to be formalized depends on the situation and need not be determined in advance. In this way the strict syntactic structure of formal proofs is slackened, compensating for the imperfection of our brains in regard to processing syntactic information.

Further, certain informal proof methods will often be described by a more or less clear reference to so-called background knowledge, and not actually carried out. This method has proven itself to be sufficiently reliable. As a matter of fact, apart from specific cases it has not yet been bettered by any of the existing automatic proof machines. Let us present a very simple example of an informal proof in a language $\mathcal{L}$ for natural numbers that along with $0, 1, +, \cdot$ contains the symbol $\mid$ for divisibility, defined by $m \mid n \Leftrightarrow \exists k\, m \cdot k = n$. In addition, let $\mathcal{L}$ contain a symbol f for some given function from $\mathbb{N}$ to $\mathbb{N}_+$. We need no closer information on this function, but we shall write f_i for $\mathsf{f}(i)$ in the example.

Example 1. We want to prove $\forall n(\exists x{>}0)(\forall i{\leqslant}n)\mathsf{f}_i \mid x$, i.e., $\mathsf{f}_0, \ldots, \mathsf{f}_n$ have a common multiple (in $\mathbb{N}_+$). The proof proceeds by induction on n. Here we focus solely on $X, (\exists x{>}0)(\forall i{\leqslant}n)\mathsf{f}_i \mid x \vDash (\exists x{>}0)(\forall i{\leqslant}n{+}1)\mathsf{f}_i \mid x$, the induction step. X represents our prior knowledge about familiar properties of divisibility. Informally we reason as follows: Assume $(\exists x{>}0)(\forall i{\leqslant}n)\mathsf{f}_i \mid x$ and choose any such x. Then $x \cdot \mathsf{f}_{n+1}$ is obviously a common multiple of $\mathsf{f}_0, \ldots, \mathsf{f}_{n+1}$, hence $(\exists x{>}0)(\forall i{\leqslant}n{+}1)\mathsf{f}_i \mid x$. To argue here formally like a proof machine, we start from $(\forall i{\leqslant}n)\mathsf{f}_i \mid x \vDash (\forall i{\leqslant}n{+}1)\mathsf{f}_i \mid (x \cdot \mathsf{f}_{n+1})$. Then posterior particularization yields $X, (\forall i{\leqslant}n)\mathsf{f}_i \mid x \vDash (\exists x{>}0)(\forall i{\leqslant}n{+}1)\mathsf{f}_i \mid x$ since $x \cdot \mathsf{f}_{n+1} > 0$, and so, after anterior particularization, we get the desired $X, (\exists x{>}0)(\forall i{\leqslant}n)\mathsf{f}_i \mid x \vDash (\exists x{>}0)(\forall i{\leqslant}n{+}1)\mathsf{f}_i \mid x$. This example shows that formalizing a nearly trivial informal argument may need a lot of writing without making things more lucid for mathematicians.

Some textbooks deal with a somewhat stricter consequence relation, which we denote here by $\overset{g}{\vDash}$. The reason is that in mathematics one largely considers derivations in theories. For $X \subseteq \mathcal{L}$ and $\varphi \in \mathcal{L}$ define $X \overset{g}{\vDash} \varphi$ if $\mathcal{A} \vDash X \Rightarrow \mathcal{A} \vDash \varphi$, for all $\mathcal{L}$-structures $\mathcal{A}$. In contrast to $\vDash$, which may be called the *local* consequence relation, $\overset{g}{\vDash}$ can be considered as the *global* consequence relation since it cares only about $\mathcal{A}$, not about a concrete valuation w in $\mathcal{A}$ as does $\vDash$.

Let us collect a few properties of $\overset{g}{\vDash}$. Obviously, $X \vDash \varphi$ implies $X \overset{g}{\vDash} \varphi$, but the converse does not hold in general. For example, $x = y \overset{g}{\vDash} \forall xy\, x = y$, but $x = y \nvDash \forall xy\, x = y$. By (d) from page 79, $X \vDash \varphi \Rightarrow X \vDash \varphi^g$ holds in general only if the free variables of φ do not occur free in X, while $X \overset{g}{\vDash} \varphi \Rightarrow X \overset{g}{\vDash} \varphi^g$ (hence $\varphi \overset{g}{\vDash} \varphi^g$) holds unrestrictedly. A reduction of $\overset{g}{\vDash}$ to $\vDash$ is provided by the following equivalence, which easily follows from $\mathcal{M} \vDash X^g \;\Leftrightarrow\; \mathcal{A} \vDash X^g$, for each model $\mathcal{M} = (\mathcal{A}, w)$:

(1) $\quad X \overset{g}{\vDash} \varphi \;\Leftrightarrow\; X^g \vDash \varphi.$

Because of $S^g = S$ for sets of sentences S, we clearly obtain from (1)

(2) $\quad S \overset{g}{\vDash} \varphi \;\Leftrightarrow\; S \vDash \varphi \quad (S \subseteq \mathcal{L}^0).$

In particular, $\overset{g}{\vDash} \varphi \Leftrightarrow\; \vDash \varphi$. Thus, a distinction between $\vDash$ and $\overset{g}{\vDash}$ is apparent only when premises are involved that are not sentences. In this case the relation $\overset{g}{\vDash}$ must be treated with the utmost care. Neither the rule of case distinction $\dfrac{X, \alpha \overset{g}{\vDash} \beta \mid X, \neg\alpha \overset{g}{\vDash} \beta}{X \overset{g}{\vDash} \beta}$ nor the deduction theorem $\dfrac{X, \alpha \overset{g}{\vDash} \beta}{X \overset{g}{\vDash} \alpha \to \beta}$ is unrestrictedly correct. For example $x = y \overset{g}{\vDash} \forall xy\, x = y$, but it is false that $\overset{g}{\vDash} x = y \to \forall xy\, x = y$. This means that the deduction theorem fails to hold for the relation $\overset{g}{\vDash}$. It holds only under certain restrictions.

One of the reasons for our preference of $\vDash$ over $\overset{g}{\vDash}$ is that $\vDash$ extends the propositional consequence relation conservatively, so that features such as the deduction theorem carry over unrestrictedly, while this is not the case for $\overset{g}{\vDash}$. It should also be said that $\overset{g}{\vDash}$ does not reflect the actual procedures of natural deduction in which formulas with free variables are frequently used also in deductions of sentences from sentences, for instance in Example 1.

We now make more precise the notion of a formalized theory in $\mathcal{L}$, where it is useful to think of the examples in **2.3**, such as group theory. Again, the definitions by different authors may look somewhat differently.

Definition. An *elementary theory* or *first-order theory* in $\mathcal{L}$, also termed an *$\mathcal{L}$-theory*, is a set of sentences $T \subseteq \mathcal{L}^0$ *deductively closed in* $\mathcal{L}^0$, i.e., $T \vDash \alpha \;\Leftrightarrow\; \alpha \in T$, for all $\alpha \in \mathcal{L}^0$. If $\alpha \in T$ then we say that α *is valid* or *true* or *holds in* T, or α *is a theorem* of T. The extralogical symbols of $\mathcal{L}$ are called the *symbols of* T. If $T \subseteq T'$ then T is called a *subtheory* of T', and T' an *extension* of T. An $\mathcal{L}$-structure $\mathcal{A}$ such that $\mathcal{A} \vDash T$ is also termed a *model of* T, briefly a *T-model*. $\mathrm{Md}\, T$ denotes the class of all models of T in this sense; $\mathrm{Md}\, T$ consist of $\mathcal{L}$-structures only.

For instance, $\{\alpha \in \mathcal{L}^0 \mid X \vDash \alpha\}$ is a theory for any set $X \subseteq \mathcal{L}$, since $\vDash$ is transitive. A theory T in $\mathcal{L}$ satisfies $T \vDash \varphi \;\Leftrightarrow\; \mathcal{A} \vDash \varphi$ for all $\mathcal{A} \vDash T$, where $\varphi \in \mathcal{L}$ is any formula. Important is also $T \vDash \varphi \;\Leftrightarrow\; T \vDash \varphi^g$. These readily confirmed facts should be taken in and remembered, since they are constantly used. Different authors may use different definitions for a theory. For example, they may not demand that theories contain sentences only, as we do. Conventions of this type each have their advantages and disadvantages. Proofs regarding theories are always adaptable enough to accommodate small modifications of the definition. Using the definition given above we set the following

Convention. In talking of the *theory S*, where S is a set of sentences, we always mean the theory determined by S, that is, $\{\alpha \in \mathcal{L}^0 \mid S \vDash \alpha\}$. A set $X \subseteq \mathcal{L}$ is called an *axiom system* for T whenever $T = \{\alpha \in \mathcal{L}^0 \mid X^g \vDash \alpha\}$, i.e., we tacitly generalize all possibly open formulas in X. We have always to think of free variables occurring in axioms as being generalized.

Thus, axioms of a theory are always sentences. But we conform to standard practice of writing long axioms as formulas. We will later consider extensive axiom systems (in particular, for arithmetic and set theory) whose axioms are partly written as open formulas just for economy.

There exists a smallest theory in $\mathcal{L}$, namely the set Taut $(= \mathit{Taut}_{\mathcal{L}})$ of all generally valid sentences in $\mathcal{L}$, also called the “logical” theory. An axiom system for Taut is the empty set of axioms. There is also a largest theory: the set $\mathcal{L}^0$ *of all* sentences, the *inconsistent* theory, which possesses no models. All remaining theories are called *satisfiable* or *consistent*.[6] Moreover, the intersection $T = \bigcap_{i\in I} T_i$ of a nonempty family of theories T_i is in turn a theory: if $T \vDash \alpha \in \mathcal{L}^0$ then clearly $T_i \vDash \alpha$ and so $\alpha \in T_i$ for each $i \in I$, hence $\alpha \in T$ as well. In this book T and T', with or without indices, exclusively denote theories.

For $T \subseteq \mathcal{L}^0$ and $\alpha \in \mathcal{L}^0$ let $T + \alpha$ denote the smallest theory that extends T and contains α. Similarly let $T + S$ for $S \subseteq \mathcal{L}^0$ be the smallest theory containing $T \cup S$. If S is finite then $T' = T + S = T + \bigwedge S$ is called a *finite extension of* T. Here $\bigwedge S$ denotes the conjunction of all sentences in S. A sentence α is termed *compatible* or *consistent* with T if $T + \alpha$ is

[6] *Consistent* mostly refers to a logic calculus, e.g., the calculus in **3.1**. However, it will be shown in **3.2** that consistency and satisfiability of a theory coincide, thus justifying the word’s ambiguous use.

satisfiable, and *refutable in* T if $T+\neg\alpha$ is satisfiable. Thus, the theory T_F of fields is *compatible* with the sentence $1+1=0$. Equivalently, $1+1\neq 0$ is *refutable* in T_F, since the 2-element field satisfies $1+1=0$.

If both α and $\neg\alpha$ are compatible with T then the sentence α is termed *independent* of T. The classic example is the independence of the parallel axiom from the remaining axioms of Euclidean plane geometry, which define *absolute* geometry. Much more difficult is the independence proof of the continuum hypothesis from the axioms for set theory. These axioms are presented and discussed in **3.4**.

At this point we introduce another important concept; $\alpha,\beta\in\mathcal{L}$ are said to be *equivalent in* or *modulo* T, $\alpha\equiv_T\beta$, if $\alpha\equiv_{\mathcal{A}}\beta$ for all $\mathcal{A}\vDash T$. Being an intersection of congruences, $\equiv_T$ is itself a congruence and hence satisfies the replacement theorem. This will henceforth be used without mention, as will the obvious equivalence of $\alpha\equiv_T\beta$, $T\vDash\alpha\leftrightarrow\beta$, and of $T\vDash(\alpha\leftrightarrow\beta)^g$. A suggestive writing of $\alpha\equiv_T\beta$ would also be $\alpha=_T\beta$.

Example 2. Let T_G be as on p. 65. Claim: $x\circ x=x\equiv_{T_G}x=e$. The only tricky proof step is $T_G\vDash x\circ x=x\to x=e$. Let $x\circ x=x$ and choose some y with $x\circ y=e$. The claim then follows from $x=x\circ e=x\circ x\circ y=x\circ y=e$. A strict formal proof of the latter uses anterior particularization.

Another important congruence is term equivalence. Call terms s,t *equivalent* modulo (or in) T, in symbols $s\approx_T t$, if $T\vDash s=t$, that is, $\mathcal{A}\vDash s=t\,[w]$ for all $\mathcal{A}\vDash T$ and $w\colon Var\to A$. For instance, in $T=T_G^=$, $(x\circ y)^{-1}=y^{-1}\circ x^{-1}$ is easily provable, so that $(x\circ y)^{-1}\approx_T y^{-1}\circ x^{-1}$. Another example: in the theory of fields, each term is equivalent to a polynomial in several variables with integer coefficients.

If all axioms of a theory T are $\forall$-sentences then T is called a *universal* or $\forall$-theory. Examples are partial orders, orders, rings, lattices, and Boolean algebras. For such a theory, $\mathrm{Md}\,T$ is closed with respect to substructures, which means $\mathcal{A}\subseteq\mathcal{B}\vDash T\Rightarrow\mathcal{A}\vDash T$. This follows at once from Corollary 3.3. Conversely, a theory closed with respect to substructures is necessarily a universal one, as will turn out in **5.4**. $\forall$-theories are further classified. The most important subclasses are equational, quasi-equational, and universal Horn theories, all of which will be considered to some extent in later chapters. Besides $\forall$-theories, the $\forall\exists$-theories (those having $\forall\exists$-sentences as axioms) are of particular interest for mathematics. More about all these theories will be said in **5.4**.

Theories are frequently given by structures or classes of structures. The elementary theory $\mathit{Th}\mathcal{A}$ and the theory $\mathit{Th}\boldsymbol{K}$ of a nonempty class $\boldsymbol{K}$ of structures are defined respectively by

$$\mathit{Th}\mathcal{A} := \{\alpha \in \mathcal{L}^0 \mid \mathcal{A} \vDash \alpha\}, \quad \mathit{Th}\boldsymbol{K} := \bigcap\{\mathit{Th}\mathcal{A} \mid \mathcal{A} \in \boldsymbol{K}\}.$$

It is easily seen that $\mathit{Th}\mathcal{A}$ and $\mathit{Th}\boldsymbol{K}$ are theories in the precise sense defined above. Instead of $\alpha \in \mathit{Th}\boldsymbol{K}$ one often writes $\boldsymbol{K} \vDash \alpha$. In general, Md $\mathit{Th}\boldsymbol{K}$ is larger than $\boldsymbol{K}$, as we shall see.

One easily confirms that the set of formulas breaks up modulo T (more precisely, modulo $\equiv_T$) into equivalence classes; their totality is denoted by $B_\omega T$. Based on these we can define in a natural manner operations $\wedge, \vee, \neg$. For instance, $\bar{\alpha} \wedge \bar{\beta} = \overline{\alpha \wedge \beta}$, where $\bar{\varphi}$ denotes the equivalence class to which φ belongs. One shows easily that $B_\omega T$ forms a Boolean algebra with respect to $\wedge, \vee, \neg$. For every n, the set $B_n T$ of all $\bar{\varphi}$ in $B_\omega T$ such that the free variables of φ belong to Var_n $(= \{\boldsymbol{v}_0, \dots, \boldsymbol{v}_{n-1}\})$ is a subalgebra of $B_\omega T$. Note that $B_0 T$ is isomorphic to the Boolean algebra of all sentences modulo $\equiv_T$, also called the *Tarski–Lindenbaum algebra* of T. The significance of the Boolean algebras $B_n T$ is revealed only in the somewhat higher reaches of model theory, and they are therefore mentioned only incidentally.

Exercises

1. Suppose $x \notin \mathit{free}\, X$ and c is not in X, α. Prove the equivalence of

$$\text{(i) } X \vDash \alpha, \qquad \text{(ii) } X \vDash \forall x\alpha, \qquad \text{(iii) } X \vDash \alpha\,\tfrac{c}{x}.$$

This holds then in particular if X is the axiom system of a theory or itself a theory. Then $x \notin \mathit{free}\, X$ is trivially satisfied.

2. Let S be a set of sentences, α and β formulas, $x \notin \mathit{free}\,\beta$, and let c be a constant not occurring in S, α, β. Show that

$$S \vDash \alpha\,\tfrac{c}{x} \rightarrow \beta \;\Leftrightarrow\; S \vDash \exists x\alpha \rightarrow \beta.$$

3. Verify for all $\alpha, \beta \in \mathcal{L}^0$ that $\beta \in T + \alpha \;\Leftrightarrow\; \alpha \rightarrow \beta \in T$.

4. Let $T \subseteq \mathcal{L}$ be a theory, $\mathcal{L}_0 \subseteq \mathcal{L}$, and $T_0 := T \cap \mathcal{L}_0$. Prove that T_0 is also a theory (the so-called *reduct theory* in the language $\mathcal{L}_0$).

2.6 Explicit Definitions—Language Expansions

The deductive development of a theory, be it given by an axiom system or a single structure or classes of those, nearly always goes hand in hand with expansions of the language carried out step by step. For example, in developing elementary number theory in the language $\mathcal{L}(0,1,+,\cdot)$, the introduction of the divisibility relation by means of the (explicit) definition $x\,|\,y \leftrightarrow \exists z\, x\cdot z = y$ has certainly advantages not only for purely technical reasons. This and similar examples motivate the following

Definition I. Let r be an n-ary relation symbol not occurring in $\mathcal{L}$. An *explicit definition of* r *in* $\mathcal{L}$ is to mean a formula of the shape

$$\eta_r: \quad r\vec{x} \leftrightarrow \delta(\vec{x})$$

with $\delta(\vec{x}) \in \mathcal{L}$ and distinct variables in $\vec{x}$, called the *defining formula.* For a theory $T \subseteq \mathcal{L}$, the extension $T_r := T + \eta_r^g$ is then called a *definitorial extension* (or *expansion*) of T by r, more precisely, by η_r.

T_r is a theory in $\mathcal{L}[r]$, the language resulting from $\mathcal{L}$ by adjoining the symbol r. It will turn out that T_r is a *conservative extension* of T, which, in the general case, means a theory $T' \supseteq T$ in $\mathcal{L}' \supseteq \mathcal{L}$ such that $T' \cap \mathcal{L} = T$. Thus, T_r contains exactly the same $\mathcal{L}$-sentences as does T. In this sense, T_r is a harmless extension of T. Our claim constitutes part of Theorem 6.1. For $\varphi \in \mathcal{L}[r]$ define the *reduced formula* $\varphi^{rd} \in \mathcal{L}$ as follows: Starting from the left, replace every prime formula $r\vec{t}$ occurring in φ by $\delta_{\vec{x}}(\vec{t}\,)$. Clearly, $\varphi^{rd} = \varphi$, provided r does not appear in φ.

Theorem 6.1 (Elimination theorem). *Let $T_r \subseteq \mathcal{L}[r]$ be a definitorial extension of the theory $T \subseteq \mathcal{L}^0$ by the explicit definition η_r. Then for all formulas $\varphi \in \mathcal{L}[r]$ holds the equivalence*

$$(*) \quad T_r \vDash \varphi \;\Leftrightarrow\; T \vDash \varphi^{rd}.$$

For $\varphi \in \mathcal{L}$ we get in particular $T_r \vDash \varphi \Leftrightarrow T \vDash \varphi$ (since $\varphi^{rd} = \varphi$). Hence, T_r is a conservative extension of T, i.e., $\alpha \in T_r \Leftrightarrow \alpha \in T$, for all $\alpha \in \mathcal{L}^0$.

Proof. Each $\mathcal{A} \vDash T$ is expandable to a model $\mathcal{A}' \vDash T_r$ with the same domain, setting $r^{\mathcal{A}'}\vec{a} :\Leftrightarrow \mathcal{A} \vDash \delta\,[\vec{a}]$ $(\vec{a} \in A^n)$. Since $r\vec{t} \equiv_{T_r} \delta(\vec{t}\,)$ for any $\vec{t}$, we obtain $\varphi \equiv_{T_r} \varphi^{rd}$ for all $\varphi \in \mathcal{L}[r]$ by the replacement theorem. Thus, $(*)$ follows from

$$\begin{aligned} T_r \vDash \varphi &\Leftrightarrow \mathcal{A}' \vDash \varphi \text{ for all } \mathcal{A} \vDash T && (\mathrm{Md}\, T_r = \{\mathcal{A}' \mid \mathcal{A} \vDash T\}) \\ &\Leftrightarrow \mathcal{A}' \vDash \varphi^{rd} \text{ for all } \mathcal{A} \vDash T && (\text{because } \varphi \equiv_{T_r} \varphi^{rd}) \\ &\Leftrightarrow \mathcal{A} \vDash \varphi^{rd} \text{ for all } \mathcal{A} \vDash T && (\text{Theorem 3.1}) \\ &\Leftrightarrow T \vDash \varphi^{rd}. \quad ❑ \end{aligned}$$

Operation symbols and constants can be similarly introduced, though in this case there are certain conditions to observe. For instance, in T_G (see page 65) the operation ${}^{-1}$ is defined by $\eta:\ y = x^{-1} \leftrightarrow x \circ y = e$. This definition is legitimate, since $T_G \vDash \forall x \exists! y\, x \circ y = e$; Exercise 3. Only this requirement (which by the way is a logical consequence of η) ensures that $T_G + \eta^g$ is a conservative extension of T_G. We therefore extend Definition I as follows, keeping in mind that to the end of this section constant symbols are to be counted among the operation symbols.

Definition II. An *explicit definition* of an n-ary operation symbol f not occurring in $\mathcal{L}$ is a formula of the form

$$\eta_f: \quad y = f\vec{x} \leftrightarrow \delta(\vec{x}, y) \qquad (\delta \in \mathcal{L} \text{ and } y, x_1, \dots, x_n \text{ distinct}).$$

η_f is called *legitimate* in $T \subseteq \mathcal{L}$ if $T \vDash \forall \vec{x}\, \exists! y \delta$, and $T_f := T + \eta_f^g$ is then called a *definitorial extension by* f, more precisely by η_f. In the case $n = 0$ we write c for f and speak of an *explicit definition of the constant symbol* c. Written more suggestively $y = c \leftrightarrow \delta(y)$.

Some of the free variables of δ are often not explicitly named, and thus downgraded to parameter variables. More on this will be said in the discussion of the axioms for set theory in **3.4**. The elimination theorem is proved in almost exactly the same way as above, provided η_f is legitimate in T. The reduced formula φ^{rd} is defined correspondingly. For a constant c ($n = 0$ in Definition II), let $\varphi^{rd} := \exists z(\varphi \frac{z}{c} \wedge \delta \frac{z}{y})$, where $\varphi \frac{z}{c}$ denotes the result of replacing c in φ by z ($\notin var\, \varphi$). Now let $n > 0$. If f does not appear in φ, set $\varphi^{rd} = \varphi$. Otherwise, looking at the first occurrence of f in φ from the left, we certainly may write $\varphi = \varphi_0 \frac{f\vec{t}}{y}$ for appropriate φ_0, $\vec{t}$, and $y \notin var\, \varphi$. Clearly, $\varphi \equiv_{T_f} \exists y(\varphi_0 \wedge y = f\vec{t}) \equiv_{T_f} \varphi_1$, with $\varphi_1 := \exists y(\varphi_0 \wedge \delta_f(\vec{t}, y))$. If f still occurs in φ_1 then repeat this procedure, which ends in, say, m steps in a formula φ_m that no longer contains f. Then put $\varphi^{rd} := \varphi_m$.

Frequently, operation symbols f are introduced in more or less strictly formalized theories by definitions of the form

$$(*) \quad f\vec{x} := t(\vec{x}),$$

where of course f does not occur in the term $t(\vec{x})$. This procedure is in fact subsumed by Definition II, because the former is nothing more than a definitorial extension of T with the explicit definition

$$\eta_f: \ y = f\vec{x} \leftrightarrow y = t(\vec{x}).$$

This definition is legitimate, since $\forall\vec{x}\,\exists!y\,y = t(\vec{x})$ is a tautology. It can readily be shown that η_f^g is logically equivalent to $\forall\vec{x}\,f\vec{x} = t(\vec{x})$. Hence, $(*)$ can indeed be regarded as a kind of an informative abbreviation of a legitimate explicit definition with the defining formula $y = t(\vec{x})$.

Remark 1. Instead of introducing new operation symbols, so-called *iota-terms* from [HB] could be used. For any formula $\varphi = \varphi(\vec{x}, y)$ in a given language, let $\iota y\varphi$ be a term in which y appears as a variable *bound by* ι. Whenever $T \vDash \forall\vec{x}\exists!y\varphi$, then T is extended by the axiom $\forall\vec{x}\forall y[y = \iota y\varphi(\vec{x}, y) \leftrightarrow \varphi(\vec{x}, y)]$, so that $\iota y\varphi(\vec{x}, y)$ so to speak stands for the function term $f\vec{x}$, which could have been introduced by an explicit definition. We mention that a definitorial language expansion is not a necessity. In principle, formulas of the expanded language can always be understood as abbreviations in the original language. This is in some presentations the actual procedure, though our imagination prefers additional notions over long sentences that would arise if we were to stick to a minimal set of basic notions.

Definitions I and II can be unified in a more general declaration. Let T, T' be theories in the languages $\mathcal{L}$, $\mathcal{L}'$, respectively. Then T' is called a *definitorial extension* (or *expansion*) of T whenever $T' = T + \Delta$ for some list Δ of explicit definitions of new symbols legitimate in T, given in terms of those of T (here *legitimate* refers to operation symbols and constants only). Δ need not be finite, but in most cases it is finite. A reduced formula $\varphi^{rd} \in \mathcal{L}$ is stepwise constructed as above, for every $\varphi \in \mathcal{L}'$. In this way the somewhat long-winded proof of the following theorem is reduced each time to the case of an extension by a single symbol:

Theorem 6.2 (General elimination theorem). *Let T' be a definitorial extension of T. Then $\alpha \in T' \Leftrightarrow \alpha^{rd} \in T$. In particular, $\alpha \in T' \Leftrightarrow \alpha \in T$ whenever $\alpha \in \mathcal{L}$, i.e., T' is a conservative extension of T.*

A relation or operation symbol s occurring in $T \subseteq \mathcal{L}$ is termed *explicitly definable in* T if T contains an explicit definition of s whose defining formula belongs to $\mathcal{L}_0$, the language of symbols of T without s. For example, in the theory T_G of groups the constant e is explicitly defined by $x = e \leftrightarrow x \circ x = x$; Example 2 page 83. Another example is presented in Exercise 3. In such a case each model of $T_0 := T \cap \mathcal{L}_0$ can be expanded in only one way to a T-model. If this special condition is fulfilled then s is said to be *implicitly definable* in T. This could also be stated as follows: if T' is distinct from T only in that the symbol s is everywhere replaced by a new symbol s', then either $T \cup T' \vDash \forall\vec{x}(\mathsf{s}\vec{x} \leftrightarrow \mathsf{s}'\vec{x})$ or $T \cup T' \vDash \forall\vec{x}(\mathsf{s}\vec{x} = \mathsf{s}'\vec{x})$, depending on whether s, s' are relation or operation symbols. It is highly interesting that this kind of definability is already sufficient for the explicit definability of s in T. But we will go without the proof and only quote the following theorem.

Beth's definability theorem. *A relation or operation symbol implicitly definable in a theory T is also explicitly definable in T.*

Definitorial expansions of a language should be conscientiously distinguished from expansions of languages that arise from the introduction of so-called *Skolem functions*. These are useful for many purposes and are therefore briefly described.

Skolem normal forms. According to Theorem 4.2, every formula α can be converted into an equivalent PNF, $\alpha \equiv \mathtt{Q}_1 x_1 \cdots \mathtt{Q}_k x_k \alpha'$, where α' is open. Obviously then $\neg\alpha \equiv \overline{\mathtt{Q}}_1 x_1 \cdots \overline{\mathtt{Q}}_k x_k \neg\alpha'$, where $\overline{\forall} = \exists$ and $\overline{\exists} = \forall$. Because $\vDash \alpha$ if and only if $\neg\alpha$ is unsatisfiable, the decision problem for general validity can first of all be reduced to the satisfiability problem for formulas in PNF. Using Theorem 6.3 below, the latter—at the cost of introducing new operation symbols—is then completely reduced to the satisfiability problem for $\forall$-formulas.

Call formulas α and β *satisfiably equivalent* if both are satisfiable (not necessarily in the same model), or both are unsatisfiable. We construct for every formula, which w.l.o.g. is assumed to be given in prenex form $\alpha = \mathtt{Q}_1 x_1 \cdots \mathtt{Q}_k x_k \beta$, a satisfiably equivalent $\forall$-formula $\hat{\alpha}$ with additional operation symbols such that $\mathit{free}\,\hat{\alpha} = \mathit{free}\,\alpha$. The construction of $\hat{\alpha}$ will be completed after m steps, where m is the number of $\exists$-quantifiers among the $\mathtt{Q}_1, \dots, \mathtt{Q}_k$. Take $\alpha = \alpha_0$ and α_i to be already constructed. If α_i is already an $\forall$-formula let $\hat{\alpha} = \alpha_i$. Otherwise α_i has the form $\forall x_1 \cdots \forall x_n \exists y \beta_i$ for some $n \geqslant 0$. With an n-ary operation symbol f (which is a constant in case n=0) not yet used let $\alpha_{i+1} = \forall\vec{x}\beta_i \frac{f\vec{x}}{y}$. Thus, after m steps an $\forall$-formula $\hat{\alpha}$ is obtained such that $\mathit{free}\,\hat{\alpha} = \mathit{free}\,\alpha$; this formula $\hat{\alpha}$ is called a *Skolem normal form* (SNF) of α.

Example 1. If α is the formula $\forall x\exists y\, x < y$ then $\hat{\alpha}$ is just $\forall x\, x < fx$.
For $\alpha = \exists x\forall y\, x \cdot y = y$ we have $\hat{\alpha} = \forall y\, c \cdot y = y$.
If $\alpha = \forall x\forall y\exists z(x < z \land y < z)$ then $\hat{\alpha} = \forall x\forall y(x < fxy \land y < fxy)$.

Theorem 6.3. *Let $\hat{\alpha}$ be a Skolem normal form for the formula α. Then*

(a) $\hat{\alpha} \vDash \alpha$, (b) *$\alpha$ is satisfiably equivalent to $\hat{\alpha}$.*

Proof. (a): It suffices to show that $\alpha_{i+1} \vDash \alpha_i$ for each of the described construction steps. $\beta_i \frac{f\vec{x}}{y} \vDash \exists y\beta_i$ implies $\alpha_{i+1} = \forall\vec{x}\beta_i \frac{f\vec{x}}{y} \vDash \forall\vec{x}\,\exists y\beta_i = \alpha_i$, by (c) and (d) in **2.5**. (b): If $\hat{\alpha}$ is satisfiable then by (a) so too is α. Conversely, suppose $\mathcal{A} \vDash \forall\vec{x}\,\exists y\beta_i(\vec{x}, y, \vec{z})\,[\vec{c}]$. For each $\vec{a} \in A^n$ we choose some $b \in A$ such that $\mathcal{A} \vDash \beta\,[\vec{a}, b, \vec{c}]$ (which is possible in view of the axiom of choice AC) and expand $\mathcal{A}$ to $\mathcal{A}'$ by setting $f^{\mathcal{A}'}\vec{a} = b$ for the new operation symbol. Then evidently $\mathcal{A}' \vDash \alpha_{i+1}\,[\vec{c}]$. Thus, we finally obtain a model for $\hat{\alpha}$ that expands the initial model. ❑

Now, for each α, a tautologically equivalent $\exists$-formula $\check{\alpha}$ is gained as well (that is, $\vDash \alpha \Leftrightarrow\ \vDash \check{\alpha}$). By the above theorem, we first produce for $\beta = \neg\alpha$ a satisfiably equivalent SNF $\hat{\beta}$ and put $\check{\alpha} := \neg\hat{\beta}$. Then indeed $\vDash \alpha \ \Leftrightarrow\ \vDash \check{\alpha}$, because

$$\vDash \alpha \ \Leftrightarrow\ \beta \text{ unsatisfiable } \Leftrightarrow\ \hat{\beta} \text{ unsatisfiable } \Leftrightarrow\ \vDash \check{\alpha}.$$

Example 2. For $\alpha := \exists x\forall y(ry \to rx)$ we have $\neg\alpha \equiv \beta := \forall x\exists y(ry \land \neg rx)$ and $\hat{\beta} = \forall x(rfx \land \neg rx)$. Thus, $\check{\alpha} = \neg\hat{\beta} \equiv \exists x(rfx \to rx)$. The last formula is a tautology. Indeed, if $r^{\mathcal{A}} \neq \emptyset$ then clearly $\mathcal{A} \vDash \exists x(rfx \to rx)$. But the same holds if $r^{\mathcal{A}} = \emptyset$, for then never $\mathcal{A} \vDash rfx$. Thus, $\check{\alpha}$ and hence also α is a tautology, which is not at all obvious after a first glance at α. This shows how useful Skolem normal forms can be for discovering tautologies.

Remark 2. There are many applications of Skolem normal forms, mainly in model theory and in logic programming. For instance, Exercise 5 permits one to reduce the satisfiability problem of an arbitrary first-order formula set to a set of $\forall$-formulas (at the cost of adjoining new function symbols). Moreover, a set X of $\forall$-formulas is satisfiably equivalent to a set X' of open formulas as will be shown in **4.1**, and this problem can be reduced completely to the satisfiability of a suitable set of propositional formulas, see also Remark 1 in **4.1**. The examples of applications of the propositional compactness theorem in **1.5** give a certain feeling for how to proceed in this way.

Exercises

1. Suppose that T_f results from a consistent theory T by adjoining an explicit definition η for f and let α^{rd} be constructed as explained in the text. Show that T_f is a conservative extension of T if and only if η is a legitimate explicit definition.

2. Let $\mathtt{S}\colon n \mapsto n+1$ denote the successor function in $\mathcal{N} = (\mathbb{N}, 0, \mathtt{S}, +, \cdot)$. Show that $Th\mathcal{N}$ is a definitorial extension of $Th(\mathbb{N}, \mathtt{S}, \cdot)$; in other words, 0 and $+$ are explicitly definable by $\mathtt{S}$ and $\cdot$ in $\mathcal{N}$.

3. Prove that $\eta : y = x^{-1} \leftrightarrow x \circ y = e$ is a legitimate explicit definition in T_G (it suffices to prove $T_G \vDash x \circ y = x \circ z \to y = z$). Show in addition that $T_G^{=} = T_G + \eta$. Thus, $T_G^{=}$ is a definitorial and hence a conservative extension of T_G. In this sense, the theories $T_G^{=}$ and T_G are equivalent formulations of the theory of groups.

4. As is well known, the natural $<$-relation of $\mathbb{N}$ is explicitly definable in $(\mathbb{N}, 0, +)$, for instance, by $x < y \leftrightarrow (\exists z \neq 0)\, z + x = y$. Prove that the $<$-relation of $\mathbb{Z}$ is not explicitly definable in $(\mathbb{Z}, 0, +)$.

5. Construct to each $\alpha \in X$ ($\subseteq \mathcal{L}$) an SNF $\hat{\alpha}$ such that X is satisfiably equivalent to $\hat{X} = \{\hat{\alpha} \mid \alpha \in X\}$ and $\hat{X} \vDash X$, called a *Skolemization* of X. Since we do not suppose that X is countable, the function symbols introduced in $\hat{X}$ must properly be indexed.

Bibliography

[AGM] S. Abramsky, D. M. Gabbay, T. S. E. Maibaum (editors), *Handbook of Computer Science*, I–IV, Oxford Univ. Press, Vol. I, II 1992, Vol. III 1994, Vol. IV 1995.

[Ac] W. Ackermann, *Die Widerspruchsfreiheit der Allgemeinen Mengenlehre*, Mathematische Annalen 114 (1937), 305–315.

[Bar] J. Barwise (editor), *Handbook of Mathematical Logic*, North-Holland 1977.

[BF] J. Barwise, S. Feferman (editors), *Model-Theoretic Logics*, Springer 1985.

[Be1] L. D. Beklemishev, *On the classification of propositional provability logics*, Math. USSR – Izvestiya 35 (1990), 247–275.

[Be2] ______, *Iterated local reflection versus iterated consistency*, Ann. Pure Appl. Logic 75 (1995), 25–48.

[Be3] ______, *Bimodal logics for extensions of arithmetical theories*, J. Symb. Logic 61 (1996), 91–124.

[Be4] ______, *Parameter free induction and reflection*, in *Computational Logic and Proof Theory*, Lecture Notes in Computer Science 1289, Springer 1997, 103–113.

[BM] J. Bell, M. Machover, *A Course in Mathematical Logic*, North-Holland 1977.

[Ben] M. Ben-Ari, *Mathematical Logic for Computer Science*, New York 1993, 2$^{\text{nd}}$ ed. Springer 2001.

[BP] P. Benacerraf, H. Putnam (editors), *Philosophy of Mathematics, Selected Readings*, Englewood Cliffs NJ 1964, 2$^{\text{nd}}$ ed. Cambridge Univ. Press 1983, reprint 1997.

[BA] A. Berarducci, P. D'Aquino, Δ_0*-complexity of the relation* $y = \prod_{i\leqslant n} F(i)$, Ann. Pure Appl. Logic 75 (1995), 49–56.

[Bi] G. Birkhoff, *On the structure of abstract algebras*, Proceedings of the Cambridge Philosophical Society 31 (1935), 433–454.

[Boo] G. Boolos, *The Logic of Provability*, Cambridge Univ. Press 1993.

[BJ] G. Boolos, R. Jeffrey, *Computability and Logic*, 3rd ed. Cambridge Univ. Press 1989.

[BGG] E. Börger, E. Grädel, Y. Gurevich, *The Classical Decision Problem*, Springer 1997.

[Bue] S. Buechler, *Essential Stability Theory*, Springer 1996.

[Bu] S. R. Buss (editor), *Handbook of Proof Theory*, Elsevier 1998.

[Ca] G. Cantor, *Gesammelte Abhandlungen* (editor E. Zermelo), Berlin 1932, Springer 1980.

[CZ] A. Chagrov, M. Zakharyashev, *Modal Logic*, Clarendon Press 1997.

[CK] C. C. Chang, H. J. Keisler, *Model Theory*, Amsterdam 1973, 3rd ed. North-Holland 1990.

[Ch] A. Church, *A note on the Entscheidungsproblem*, J. Symb. Logic 1 (1936), 40–41, also in [Dav, 108–109].

[CM] W. Clocksin, C. Mellish, *Programming in PROLOG*, 3rd ed. Springer 1987.

[Da] D. van Dalen, *Logic and Structure*, Berlin 1980, 4th ed. Springer 2004.

[Dav] M. Davis (editor), *The Undecidable*, Raven Press 1965.

[Daw] J. W. Dawson, *Logical Dilemmas*, *The Life and Work of Kurt Gödel*, A. K. Peters 1997.

[De] O. Deiser, *Axiomatische Mengenlehre*, Springer, to appear 2010.

[Do] K. Doets, *From Logic to Logic Programming*, MIT Press 1994.

[EFT] H.-D. Ebbinghaus, J. Flum, W. Thomas, *Mathematical Logic*, New York 1984, 2nd ed. Springer 1994.

[FF] A. B. Feferman, S. Feferman, *Alfred Tarski*, *Live and Logic*, Cambridge Univ. Press 2004.

[Fe1] S. Feferman, *Arithmetization of metamathematics in a general setting*, Fund. Math. 49 (1960), 35–92.

[Fe2] ———, *In the Light of Logic*, Oxford Univ. Press 1998.

[Fel1] W. Felscher, *Berechenbarkeit*, Springer 1993.

[Fel2] ———, *Lectures on Mathematical Logic*, Vol. 1–3, Gordon & Breach 2000.

[Fi] M. Fitting, *Incompleteness in the Land of Sets*, College Publ. 2007.

[Fr] T. Franzén, *Gödel's Theorem: An Incomplete Guide to Its Use and Abuse*, A. K. Peters 2005.

[Fre] G. Frege, *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*, Halle 1879, G. Olms Verlag 1971, also in [Hei, 1–82].

[FS] H. Friedman, M. Sheard, *Elementary descent recursion and proof theory*, Ann. Pure Appl. Logic 71 (1995), 1–47.

[Ga] D. Gabbay, *Decidability results in non-classical logic III*, Israel Journal of Mathematics 10 (1971), 135–146.

[GJ] M. Garey, D. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*, Freeman 1979.

[Ge] G. Gentzen, *The Collected Papers of Gerhard Gentzen* (editor M. E. Szabo), North-Holland 1969.

[Gö1] K. Gödel, *Die Vollständigkeit der Axiome des logischen Funktionenkalküls*, Monatshefte f. Math. u. Physik 37 (1930), 349–360, also in [Gö3, Vol. I, 102–123], [Hei, 582–591].

[Gö2] ———, *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I*, Monatshefte f. Math. u. Physik 38 (1931), 173–198, also in [Gö3, Vol. I, 144–195], [Hei, 592–617], [Dav, 4–38].

[Gö3] ———, *Collected Works* (editor S. Feferman), Vol. I–V, Oxford Univ. Press, Vol. I 1986, Vol. II 1990, Vol. III 1995, Vol. IV, V 2003.

[Gor] S. N. Goryachev, *On the interpretability of some extensions of arithmetic*, Mathematical Notes 40 (1986), 561–572.

[Gr] G. Grätzer, *Universal Algebra*, New York 1968, 2$^{\text{nd}}$ ed. Springer 1979.

[HP] P. Hájek, P. Pudlák, *Metamathematics of First-Order Arithmetic*, Springer 1993.

[Hei] J. van Heijenoort (editor), *From Frege to Gödel*, Harvard Univ. Press 1967.

[He] L. Henkin, *The completeness of the first-order functional calculus*, J. Symb. Logic 14 (1949), 159–166.

[Her] J. Herbrand, *Recherches sur la théorie de la démonstration*, C. R. Soc. Sci. Lett. Varsovie, Cl. III (1930), also in [Hei, 525–581].

[HR] H. Herre, W. Rautenberg, *Das Basistheorem und einige Anwendungen in der Modelltheorie*, Wiss. Z. Humboldt-Univ., Math. Nat. R. 19 (1970), 579–583.

[HeR] B. Herrmann, W. Rautenberg, *Finite replacement and finite Hilbert-style axiomatizability*, Zeitsch. Math. Logik Grundlagen Math. 38 (1982), 327–344.

[HA] D. Hilbert, W. Ackermann, *Grundzüge der theoretischen Logik*, Berlin 1928, 6th ed. Springer 1972.

[HB] D. Hilbert, P. Bernays, *Grundlagen der Mathematik*, I, II, Berlin 1934, 1939, 2nd ed. Springer, Vol. I 1968, Vol. II 1970.

[Hi] P. Hinman, *Fundamentals of Mathematical Logic*, A. K. Peters 2005.

[Ho] W. Hodges, *Model Theory*, Cambridge Univ. Press 1993.

[Hor] A. Horn, *On sentences which are true of direct unions of algebras*, J. Symb. Logic 16 (1951), 14–21.

[Hu] T. W. Hungerford, *Algebra*, Springer 1980.

[Id] P. Idziak, *A characterization of finitely decidable congruence modular varieties*, Trans. Amer. Math. Soc. 349 (1997), 903–934.

[Ig] K. Ignatiev, *On strong provability predicates and the associated modal logics*, J. Symb. Logic 58 (1993), 249–290.

[JK] R. Jensen, C. Karp, *Primitive recursive set functions*, in *Axiomatic Set Theory, Vol. I* (editor D. Scott), Proc. Symp. Pure Math. 13, I AMS 1971, 143–167.

[Ka] R. Kaye, *Models of Peano Arithmetic*, Clarendon Press 1991.

[Ke] H. J. Keisler, *Logic with the quantifier "there exist uncountably many"*, Annals of Mathematical Logic 1 (1970), 1–93.

[Kl1] S. Kleene, *Introduction to Metamathematics*, Amsterdam 1952, 2nd ed. Wolters-Noordhoff 1988.

[Kl2] ———, *Mathematical Logic*, Wiley & Sons 1967.

[KR] I. Korec, W. Rautenberg, *Model interpretability into trees and applications*, Arch. math. Logik 17 (1976), 97–104.

[Kr] M. Kracht, *Tools and Techniques in Modal Logic*, Elsevier 1999.

[Kra] J. Krajíček, *Bounded Arithmetic, Propositional Logic, and Complexity Theory*, Cambridge Univ. Press 1995.

[KK] G. Kreisel, J.-L. Krivine, *Elements of Mathematical Logic*, North-Holland 1971.

[Ku] K. Kunen, *Set Theory, An Introduction to Independence Proofs*, North-Holland 1980.

[Le] A. Levy, *Basic Set Theory*, Springer 1979.

[Li] P. Lindström, *On extensions of elementary logic*, Theoria 35 (1969), 1–11.

[Ll] J. W. Lloyd, *Foundations of Logic Programming*, Berlin 1984, 2nd ed. Springer 1987.

[Lö] M. Löb, *Solution of a problem of Leon Henkin*, J. Symb. Logic 20 (1955), 115–118.

[MS] A. Macintyre, H. Simmons, *Gödel's diagonalization technique and related properties of theories*, Colloquium Mathematicum 28 (1973), 165–180.

[Ma] A. I. Mal'cev, *The Metamathematics of Algebraic Systems*, North-Holland 1971.

[Mal] J. Malitz, *Introduction to Mathematical Logic*, Springer 1979.

[Man] Y. I. Manin, *A Course in Mathematical Logic for Mathematicians*, New York 1977, 2nd ed. Springer 2010.

[Mar] D. Marker, *Model Theory, An Introduction*, Springer 2002.

[Mat] Y. Matiyasevich, *Hilbert's Tenth Problem*, MIT Press 1993.

[MV] R. McKenzie, M. Valeriote, *The Structure of Decidable Locally Finite Varieties*, Progress in Mathematics 79, Birkhäuser 1989.

[Me] E. Mendelson, *Introduction to Mathematical Logic*, Princeton 1964, 4th ed. Chapman & Hall 1997.

[Mo] D. Monk, *Mathematical Logic*, Springer 1976.

[Moo] G. H. Moore, *The emergence of first-order logic*, in *History and Philosophy of Modern Mathematics* (editors W. Aspray, P. Kitcher), University of Minnesota Press 1988, 95–135.

[ML] G. Müller, W. Lenski (editors), *The Ω-Bibliography of Mathematical Logic*, Springer 1987.

[Po] W. Pohlers, *Proof Theory, An Introduction*, Lecture Notes in Mathematics 1407, Springer 1989.

[Pz] B. Poizat, *A Course in Model Theory*, Springer 2000.

[Pr] M. Presburger, *Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt*, Congrès des Mathématiciens des Pays Slaves 1 (1930), 92–101.

[RS] H. Rasiowa, R. Sikorski, *The Mathematics of Metamathematics*, Warschau 1963, 3rd ed. Polish Scientific Publ. 1970.

[Ra1] W. Rautenberg, *Klassische und Nichtklassische Aussagenlogik*, Vieweg 1979.

[Ra2] ______, *Modal tableau calculi and interpolation*, Journ. Phil. Logic 12 (1983), 403–423.

[Ra3] ______, *A note on completeness and maximality in propositional logic*, Reports on Mathematical Logic 21 (1987), 3–8.

[Ra4] ______, *Einführung in die mathematische Logik*, Wiesbaden 1996, 3rd ed. Vieweg+Teubner 2008.

[Ra5] ______, *Messen und Zählen, Eine einfache Konstruktion der reellen Zahlen*, Heldermann 2007.

[RZ] W. Rautenberg, M. Ziegler, *Recursive inseparability in graph theory*, Notices Amer. Math. Soc. 22 (1975), A–523.

[Ro1] A. Robinson, *Introduction to Model Theory and to the Metamathematics of Algebra*, Amsterdam 1963, 2nd ed. North-Holland 1974.

[Ro2] ______, *Non-Standard Analysis*, Amsterdam 1966, 3rd ed. North-Holland 1974.

[Rob] J. Robinson, *A machine-oriented logic based on the resolution principle*, Journal of the ACM 12 (1965), 23–41.

[Rog] H. Rogers, *Theory of Recursive Functions and Effective Computability*, New York 1967, 2nd ed. MIT Press 1988.

[Ros] J. B. Rosser, *Extensions of some theorems of Gödel and Church*, J. Symb. Logic 1 (1936), 87–91, also in [Dav, 230–235].

[Rot] P. Rothmaler, *Introduction to Model Theory*, Gordon & Breach 2000.

[Ry] C. Ryll-Nardzewki, *The role of the axiom of induction in elemenary arithmetic*, Fund. Math. 39 (1952), 239–263.

[Sa] G. Sacks, *Saturated Model Theory*, W. A. Benjamin 1972.

[Sam] G. Sambin, *An effective fixed point theorem in intuitionistic diagonalizable algebras*, Studia Logica 35 (1976), 345–361.

[Sc] U. Schöning, *Logic for Computer Scientist*, Birkhäuser 1989.

[Se] A. Selman, *Completeness of calculi for axiomatically defined classes of algebras*, Algebra Universalis 2 (1972), 20–32.

[Sh] S. Shapiro (editor), *The Oxford Handbook of Philosophy of Mathematics and Logic*, Oxford Univ. Press 2005.

[She] S. Shelah, *Classification Theory and the Number of Nonisomorphic Models*, Amsterdam 1978, 2nd ed. North-Holland 1990.

[Shoe] J. R. Shoenfield, *Mathematical Logic*, Reading Mass. 1967, A. K. Peters 2001.

[Si] W. Sieg, *Herbrand analyses*, Arch. Math. Logic 30 (1991), 409–441.

[Sm] P. Smith, *An Introduction to Gödel's Theorems*, Cambridge Univ. Press 2007.

[Smo] C. Smoryński, *Self-reference and Modal Logic*, Springer 1984.

[Smu] R. Smullyan, *Theory of Formal Systems*, Princeton Univ. Press 1961.

[So] R. Solovay, *Provability interpretation of modal logic*, Israel Journal of Mathematics 25 (1976), 287–304.

[Sz] W. Szmielew, *Elementary properties of abelian groups*, Fund. Math. 41 (1954), 203–271.

[Tak] G. Takeuti, *Proof Theory*, Amsterdam 1975, 2nd ed. Elsevier 1987.

[Ta1] A. Tarski, *Der Wahrheitsbegriff in den formalisierten Sprachen*, Studia Philosophica 1 (1936), 261–405, also in [Ta4, 152-278].

[Ta2] ———, *Introduction to Logic and to the Methodology of Deductive Sciences*, Oxford 1941, 3rd ed. Oxford Univ. Press 1965 (first edition in Polish, 1936).

[Ta3] ———, *A Decision Method for Elementary Algebra and Geometry*, Santa Monica 1948, Berkeley 1951, Paris 1967.

[Ta4] ———, *Logic, Semantics, Metamathematics* (editor J. Corcoran), Oxford 1956, 2nd ed. Hackett 1983.

[TMR] A. Tarski, A. Mostowski, R. M. Robinson, *Undecidable Theories*, North-Holland 1953.

[TV] A. Tarski, R. Vaught, *Arithmetical extensions and relational systems*, Compositio Mathematica 13 (1957), 81–102.

[Tu] A. TURING, *On computable numbers, with an application to the Entscheidungsproblem*, Proc. London Math. Soc., 2nd Ser. 42 (1937), 230–265, also in [Dav, 115–154].

[Vi1] A. VISSER, *Aspects of Diagonalization and Provability*, Dissertation, University of Utrecht 1981.

[Vi2] ———, *An overview of interpretability logic*, in *Advances in Modal Logic, Vol. 1* (editors M. KRACHT et al.), CSLI Lecture Notes 87 (1998), 307–359.

[Wae] B. L. VAN DER WAERDEN, *Algebra I*, Berlin 1930, 4th ed. Springer 1955.

[Wag] F. WAGNER, *Simple Theories*, Kluwer Academic Publ. 2000.

[Wa1] H. WANG, *From Mathematics to Philosophy*, Routlegde & Kegan Paul 1974.

[Wa2] ———, *Computer, Logic, Philosophy*, Kluwer Academic Publ. 1990.

[Wa3] ———, *A Logical Journey, From Gödel to Philosophy*, MIT Press 1997.

[WR] A. WHITEHEAD, B. RUSSELL, *Principia Mathematica*, I–III, Cambridge 1910, 1912, 1913, 2nd ed. Cambridge Univ. Press, Vol. I 1925, Vol. II, III 1927.

[Wi] A. WILKIE, *Model completeness results for expansions of the ordered field of real numbers by restricted Pfaffian functions and the exponential function*, Journal Amer. Math. Soc. 9 (1996), 1051–1094.

[WP] A. WILKIE, J. PARIS, *On the scheme of induction for bounded arithmetic formulas*, Ann. Pure Appl. Logic 35 (1987), 261–302.

[Zi] M. ZIEGLER, *Model theory of modules*, Ann. Pure Appl. Logic 26 (1984), 149–213.

Index of Terms and Names

Index of Symbols