Unification in Modal and Description Logics*

Franz Baader
TU Dresden, Germany
baader@inf.tu-dresden.de

Silvio Ghilardi
Università degli Studi di Milano, Italy
ghilardi@dsi.unimi.it

Abstract

Unification was originally introduced in automated deduction and term rewriting, but has recently also found applications in other fields. In this article, we give a survey of the results on unification obtained in two closely related, yet different, application areas of unification: description logics and modal logics.

1 Introduction

Unification, i.e., the problem of making given terms syntactically equal by replacing their variables by terms, was independently introduced in automated deduction by Robinson [57] and in term rewriting by Knuth [48]. In both areas it later turned out that certain equational axioms (such as associativity and commutativity of a binary function symbol) are hard to deal with by theorem provers and term rewriting systems. For this reason, it was proposed in both areas [55, 54] to build in such troublesome axioms into the proof process (resolution or Knuth-Bendix completion) by using special unification procedures. Instead of making terms syntactically equal, these procedures try to make them equivalent modulo a given equational theory E. We call this type of unification E-unification or equational unification. More formally, an E-unification problem is a system of term equations of the form $\{s_1 \stackrel{?}{=}_E t_1, \ldots, s_n \stackrel{?}{=}_E t_n\}$, and an E-unifier of this system is a substitution σ such that $s_1\sigma =_E t_1\sigma, \ldots, s_n\sigma =_E t_n\sigma$, where $=_E$ denotes equality modulo E.

*This paper has been published in the Logic Journal of the IGPL (2011), 19(6):705–730. http://jigpal.oxfordjournals.org/content/19/6/705.abstract

Unification theory tries to investigate the unification properties of equational theories in a systematic way [71, 46, 16, 17]. On the one hand, one can consider the decision problem, i.e., the question whether a given E-unification problem has an E-unifier or not. For the decision problem one is interested in whether it is decidable or not. For decidable problems one can then research their complexity (see [16, 17] for an overview of such results). Instead of just deciding unifiability, one often also wants to compute E-unifiers in case they exist (i.e., if the problem is E-unifiable). This is, for example, the case in the applications of E-unification in automated deduction and term rewriting mentioned above. In these applications, one is interested in computing a minimal complete set of E-unifiers, i.e., a set of E-unifiers of the given problem such that (i) every E-unifier of the problem is an instance of a unifier in this set (completeness); and (ii) two different unifiers in the set are incomparable w.r.t. instantiation (minimality). The unification type of an equational theory says how large such sets can maximally get. The theory E is unitary (finitary, infinitary) if the cardinality of a minimal complete set of E-unifiers is at most 1 (always finite; sometimes infinite). In the worst possible case (type 0), a minimal complete set does not always exist, which in particular implies that there cannot be a finite complete set. Natural examples of equational theories of unification type 0 can, for example, be found in [67, 3, 4].

In addition to the "classical" applications of unification modulo equational theories in automated deduction and term rewriting mentioned above, unification has also turned out to be of interest in other areas. For example, it can be used in the context of verification of crytographic protocols [23, 24, 47, 27, 25, 1], where the equational theory models algebraic properties of crytographic functions. In this article, we consider two closely related, yet different application areas for unification: modal logics [19] and description logics [8]. These areas are closely related since many description logics are just syntactic variants of certain modal logics. The most prominent example is the description logic $\mathcal{ALC}$, which is a syntactic variant of the basic multi-modal logic K_m [64]. Consequently, technical results obtained in one area (such as decidability and complexity of the satisfiability problem; finite model property; etc.) can be transfered to the other, and this is, of course, also true for results regarding unification. The difference between the areas is due to the fact that they have different underlying intuitions and applications, and thus different aspects are considered to be important. For example, in modal logics, axiomatizations have always been of great importance. Many modal logics were first introduced through their axiomatizations, and the corresponding Kripke semantics only came later. Also, the axiomatizations are often closer to the intuitions underlying the use of the modal logics (e.g., as a formalization of knowledge and belief of rational agents) than their Kripke semantics. In contrast, in description logics, axiomatization are only of minor relevance, whereas the set-theoretic semantics (which corresponds to the Kripke semantics in modal logics) is the main approach for defining a description logic. This semantics directly reflects

the intended use of the description logic as a formalism for defining concepts, i.e., sets of objects that share some common properties.

These differences are also reflected in the different motivations for considering unification in the two areas. In modal logics, unification has been introduced as a special case of recognizability of admissible inference rules [61]. Intuitively, this problem asks, for a given modal logic L, whether a certain inference rule can be added to its axiomatization without changing the logic. This may, for example, be interesting if one wants to make the axiom system more efficient by adding rules that can derive consequences faster, or if one wants to check whether two axiomatizations that use different rules are equivalent. Since unification is a special case of the recognizability problem, solving it can be seen as a warm-up excercise for solving the more general problem. In addition, undecidability results for unification immediately yield undecidability of the recognizability problem. However, as we will see below, unification is not only a special case of the recognizability problem. In some cases, unification algorithms that compute finite minimal complete sets of unifiers can actually be used to solve this problem. Thus, as in the applications of equational unification in theorem proving and term rewriting, it is desirable for modal logics to have unification type unitary or finitary.

In description logics, unification was introduced as a tool for recognizing redundant concept descriptions [12, 13]. Basically, the idea is the following. Assume that you have two concept descriptions C and D that you suspect to be different formalizations of the same intuitive concept. One might think that it is enough to test whether C and D are equivalent, i.e., describe the same set of objects in every interpretation. However, this disregards that the two descriptions may employ different names for primitive concepts or be modeled on different levels of granularity, where a concept name in one description actually corresponds to a complex sub-description in the other one. This problem is overcome by testing the descriptions for unifiability modulo equivalence rather than equivalence itself. In this application, the existence of a finite minimal complete set of unifiers actually does not appear to be relevant. Instead, one is interested in computing ground unifiers, i.e., unfiers that replace all variables by concept descriptions not containing variables. In fact, the description logics whose unification properties have been investigated until now are all of unification type zero, but unifiability is decidable and the decision procedures can be used to compute a ground unifier in case the input problem is unifiable. If σ is a ground substitution, then the fact that it unifies the concept descriptions C, D basically says the following. If we were to add concept definitions $X \equiv X\sigma$ for all the concept names X that were viewed as variables in the unification problem, then we would make the two descriptions equivalent w.r.t. these definitions. Of course, the knowledge engineer needs to check whether these definitions really make sense within the application domain that is modeled with these concepts.

In most cases, unification problems in modal logics and in description logics

Name	Syntax	Semantics
concept name	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
role name	r	$r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
top concept	$\top$	$\Delta^{\mathcal{I}}$
bottom concept	$\bot$	$\emptyset$
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
value restriction	$\forall r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \forall e.(d,e) \in r^{\mathcal{I}} \rightarrow e \in C^{\mathcal{I}}\}$
existential restriction	$\exists r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \exists e.(d,e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}$

Figure 1: Syntax and semantics of $\mathcal{ALC}$ concept descriptions

can actually be viewed as unification problems modulo the equational theory that axiomatizes equivalence in the respective logic. Thus, from the point of view of unification theory, the investigation of unification in modal and description logics increases our knowledge about equational unification since it provides us with new results about the unification properties of certain equational theories.

Equational unification also plays a rôle in the context of modal logics in a quite different setting. In the so-called (optimized) functional translation of modal logics into first-order logic, frame properties are translated into equational axioms, which can be dealt with by equational unification [51, 26, 37, 52, 2, 53, 66]. This kind of unification for modal logics is very different from the one mentioned before, and will not be treated in the present paper. We refer the interested reader to the cited literature.

2 Preliminaries definitions and remarks

In this section, we first explain the connection between description and modal logics on the example of the modal logic K_m and the description logic $\mathcal{ALC}$. Then, we show how unification problems in description and modal logics can be viewed as equational unification problems. We assume that the reader is familiar with the basic notions from unification theory mentioned in the introduction (exact definitions can, e.g., be found in [17]). More information about description logic, modal logic, and their connection can be found in [8, 19].

2.1 $\mathcal{ALC}$ and K_m

Let us start with defining the description logic $\mathcal{ALC}$. Assume that a countably infinite supply of *concept names*, usually denoted A and B, and of *role names*, usually denoted r and s, are available. *Concept descriptions* in $\mathcal{ALC}$ are formed

Name	Syntax	Semantics
variable	x	$V(x) \subseteq W,\ \mathcal{M} \models_w x$ iff $w \in V(x)$
modal parameter	p	$R_p \subseteq W \times W$
truth	$\top$	$\mathcal{M} \models_w \top$
falsity	$\bot$	$\mathcal{M} \not\models_w \bot$
negation	$\neg A$	$\mathcal{M} \models_w \neg A$ iff $\mathcal{M} \not\models_w A$
conjunction	$A \wedge B$	$\mathcal{M} \models_w A \wedge B$ iff $\mathcal{M} \models_w A$ and $\mathcal{M} \models_w B$
disjunction	$C \vee D$	$\mathcal{M} \models_w A \vee B$ iff $\mathcal{M} \models_w A$ or $\mathcal{M} \models_w B$
box	$\Box_p A$	$\mathcal{M} \models_w \Box_p A$ iff for all w', $R_p(w, w')$ implies $\mathcal{M} \models_{w'} A$
diamond	$\Diamond_p A$	$\mathcal{M} \models_w \Diamond_p A$ iff there is w' with $R_p(w, w')$ and $\mathcal{M} \models_{w'} A$

Figure 2: Syntax and semantics of K_m formulae

according to the following syntax rule:

$$C, D \longrightarrow A \mid \top \mid \bot \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \forall r.C \mid \exists r.C$$

where A ranges over concept names and r ranges over role names.

The semantics of $\mathcal{ALC}$ is based on *interpretations*, i.e., pairs $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a non-empty set (the *domain* of $\mathcal{I}$), and $\cdot^{\mathcal{I}}$ is the *interpretation function*, assigning to each concept name A a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to each role name r a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function is inductively extended to concept descriptions as shown in Figure 1, which also lists the names that are usually empoyed for the $\mathcal{ALC}$ constructors. Two $\mathcal{ALC}$ concept descriptions C, D are *equivalent* ($C \equiv D$) iff they are always interpreted by the same set, i.e., $C^{\mathcal{I}} = D^{\mathcal{I}}$ holds for all interpretations $\mathcal{I}$. The $\mathcal{ALC}$ concept description D *subsumes* the $\mathcal{ALC}$ concept description C ($C \sqsubseteq D$) iff $C \sqcap D \equiv C$, i.e., $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for all interpretations $\mathcal{I}$. It is well-known that subsumption and equivalence of $\mathcal{ALC}$ concept descriptions are PSpace-complete problems [69].

$\mathcal{ALC}$ is only one member of a large family of description logics. These logics can be obtained from $\mathcal{ALC}$ by disallowing certain constructors (thus obtaining *sub-Boolean* description logics, like $\mathcal{EL}$ and $\mathcal{FL}_0$, for which we consider unification in Section 4) and/or adding various combinations of additional constructors. Such additional constructors can be concept constructors, or they can be role constructors allowing to construct compound role descriptions to be used in place of role names. A special case of such a (nullary) role constructor is the universal role u, which is interpreted as $u^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

To define the basic multi-modal logic K_m for $m \geq 1$, we assume that there is a countably infinite supply of propositional variables, usually denoted x, y, z, and m modal parameter $p_1, \ldots, p_m$. Formulae of K_m are then defined using the following syntax rule:

$$A, B \longrightarrow x \mid \top \mid \bot \mid \neg A \mid A \wedge B \mid A \vee B \mid \Box_p A \mid \Diamond_p A$$

where x ranges over propositional variables and p ranges over modal parameters.

The semantics of K_m is based on *Kripke frames* and *Kripke models*. A Kripke frame is a pair $\mathsf{F} = (W, (R_{p_1}, \ldots, R_{p_m}))$ where W is a non-empty set (of states or possible worlds), and $(R_{p_1}, \ldots, R_{p_m})$ is an m-tuple of binary relations on W (transition relations or accessibility relations). A Kripke model $\mathcal{M}$ consists of such a frame F together with a valuation V, which maps each variable to a subset of W (the worlds/states in which this variable is true). In this case, we say that $\mathcal{M}$ is *based on* the frame F. The *validity* of a formula A in the world w of the Kripke model $\mathcal{M}$ ($\mathcal{M} \models_w A$) is defined inductively, as shown in the semantics column of Figure 2. The formula A is *valid in the Kripke model* $\mathcal{M}$ ($\mathcal{M} \models A$) iff it is valid in every world of this model, and it is *valid in the Kripke frame* F ($\mathsf{F} \models A$) iff it is valid in every Kripke model based on F. Finally, A is *valid in the class of Kripke frames* $\mathcal{K}$ iff it is valid in every frame belonging to this class. The set of formulae valid in the class of frames $\mathcal{K}$ is denoted by $L(\mathcal{K})$. Usually, one calls $L(\mathcal{K})$ the *modal logic* induced by the class of frames $\mathcal{K}$. The *valid formulae of* K_m are the ones that are valid in the class of all Kripke frames. Other modal logics can be obtained by imposing restrictions on the transition relations (like transitivity, symmetry, etc.) and then taking as class of frames the ones where the transition relations satisfy these restrictions. An axiomatization of the modal logic $L(\mathcal{K})$ consists of axioms and inference rules, which can be used to derive exactly the formulae that are valid in $\mathcal{K}$, i.e., the elements of $L(\mathcal{K})$. It should be noted, however, that there are also modal logics (defined directly via an axiomatization) that are not induced by Kripke frames.

In modal logics, one usually considers uni-modal logics rather than multi-modal logics, i.e., the case where $m = 1$. The reason why we have introduced the multi-modal case is that such logics are closer to description logics. As first observed by Schild [64], $\mathcal{ALC}$ is a notational variant of K_m, i.e. there is a bijective translations $C \mapsto A_C$ from $\mathcal{ALC}$ concept descriptions to K_m formulae and a bijective translation $\mathcal{I} \mapsto \mathcal{M}_\mathcal{I}$ from $\mathcal{ALC}$ interpretations to Kripke models such that $C^\mathcal{I} = \{w \mid \mathcal{M}_\mathcal{I} \models_w A_C\}$. Syntactically, concept names can simply be viewed as propositional variables and role names can be viewed as modal parameters. Then, the $\mathcal{ALC}$ interpretations $\mathcal{I}$ can obviously be translated into the Kripke model that has $\Delta^\mathcal{I}$ as set of worlds and where $\cdot^\mathcal{I}$ provides both the transition relations and the valuation of the propositional variables. With this reading, the value restriction $\forall r.C$ becomes a box operator $\Box_r C$ referring to the accessibility relation for the modal parameter r, $\exists r.C$ becomes a diamond operator $\Diamond_r C$, and of course the Boolean operations in $\mathcal{ALC}$ are translated into the corresponding Boolean operations on the modal side.

This close connection between $\mathcal{ALC}$ and K_m also implies that inference problems in these two logics can be reduced to each other. For example, for all $\mathcal{ALC}$ concept descriptions C, D, we have $C \equiv D$ iff $A_C \leftrightarrow A_D$ is valid in K_m,[1] and A_C

[1] where $\leftrightarrow$ can be expressed in the usual way using the Boolean operations $\wedge, \vee, \neg$.

is valid in K_m iff $C \equiv \top$.

2.2 Unification problems

Unification in modal and description logics will be introduced in more detail in the next two sections. Here, we only want to mention some differences and similarities between the kinds of unification problems considered in these areas, and point out the connection to equational unification.

In description logics, one considers pairs of concept descriptions C, D that one wants to make equivalent by replacing some of the concept names occurring in them by concept descriptions. Thus, some of the concept names are declared to be concept variables (which can be replaced by a substitution), whereas the others are viewed to be constants (which cannot be replaced by a substitution). A *unifier* of C, D is then a substitution σ such that $C\sigma \equiv D\sigma$. This is actually an equational unification problem since we may view concept descriptions as terms and $\equiv$ as an equational theory on these terms. For many interesting description logics, $\equiv$ can actually be axiomatized using finitely many equational axioms (if we restrict the attention to a finite set of role names). For example, equivalence in $\mathcal{ALC}$ can be axiomatized by extending the theory of Boolean algebras (for $\sqcap, \sqcup, \neg, \top, \bot$) with the following two axioms (for every role name r):

$$\forall r.(X \sqcap Y) \equiv \forall r.X \sqcap \forall r.Y \quad \text{and} \quad \forall r.\top \equiv \top.$$

In Section 4, we consider unification in the sub-Boolean fragments $\mathcal{FL}_0$ and $\mathcal{EL}$ of $\mathcal{ALC}$. The description logic $\mathcal{FL}_0$ has the constructors top concept, conjunction, and value restriction, whereas in $\mathcal{EL}$ value restriction is replaced by existential restriction. Equivalence in these two logics can be axiomatized by the following equational theories [13, 11, 72]. For $\mathcal{FL}_0$ one takes the equational axioms stating associativity, commutativity, and idempotency of $\sqcap$, the fact that $\top$ is a unit for $\sqcap$, and the additional axioms

$$\forall r.(X \sqcap Y) \equiv \forall r.X \sqcap \forall r.Y \quad \text{and} \quad \forall r.\top \equiv \top.$$

For $\mathcal{EL}$, these two axioms must be replaced by

$$\exists r.(X \sqcap Y) \sqcap \exists r.X \equiv \exists r.(X \sqcap Y).$$

Unification in modal logics is often introduced as follows: given a formula A, one wants to make this formula valid by replacing its propositional variables with appropriate formulae. Thus, for a modal logic L (which we identify with the set of formulae valid in it) and a formula A, an *L-unifier of A* is a substitution σ such that $A\sigma \in L$. At first sight, this does not match our definition of unification in description logics. In fact, a direct transfer of this definition would yield the following definition of unification in modal logic: given formulae A, B, find a

substitution σ such that $A\sigma$ and $B\sigma$ are equivalent, i.e., such that $A\sigma \leftrightarrow B\sigma$ is valid. However, since $A\sigma \leftrightarrow B\sigma$ is valid iff $(A \leftrightarrow B)\sigma$ is valid, this is actually a special case of our first definition of unification in modal logics. Conversely, we have that $A\sigma$ is valid iff $A\sigma \leftrightarrow \top\sigma$ is valid, and thus the two definitions of unification in modal logics introduced above are actually equivalent.[2]

There is, however, another subtle difference between our definitions of unification in description logics and unification in modal logics. Given concept descriptions C, D, we assume that some (but not necessarily all) of the concepts names occurring in these descriptions are variables that can be replaced by substitutions. The remaining concept names are constants in the sense that they cannot be replaced by substitutions.[3] Thus, in the terminology of unification theory [17], such a unification problem is a unification problem *with constants*. In contrast, for a given modal formula A that we want to make valid by unification, we have assumed that all its propositional variables are variables that can be replaced by substitutions. Thus, the only available constants are $\top$ and $\bot$, which are, however, theory constants (in the sense that they occur in the equations that axiomatize equivalence). Thus, the unification problems for modal logics introduced above are so-called *elementary* unification problems. As pointed out in [17], Section 3.2.2, the unification properties (unification type, decidability and complexity of unification problems) of a given equational theory may differ, depending on whether one considers elementary unification problems or unification problems with constants. Most of the work on unification in modal logics considers only elementary unification problems. In fact, unification in modal logics is usually investigated in the context of admissible inference rules, and in these rules, the presence of constants (called parameters in [61]) does not appear to be very important for the intended applications.

As also pointed out in [17] (Section 3.2.3), the unification properties of a given equational theory may differ, depending on whether one considers single equations or systems of equations. In this terminology, what we have introduced until now as unification problems are single equations: in description logics, our unification problems consist of *one* pair of concept descriptions C, D, and in modal logics of *one* formula A. In Section 4, we will actually allow unification problems in description logics to be systems of equations of the form $\{C_1 \equiv^? D_1, \ldots, C_n \equiv^? D_n\}$, which must be solved simultaneously by a unifier, i.e., a unifier σ must satisfy $C_1\sigma \equiv D_1\sigma, \ldots, C_n\sigma \equiv D_n\sigma$. However, for the description logics $\mathcal{ALC}$, $\mathcal{EL}$, and $\mathcal{FL}_0$, this is not really a generalization since systems of equations can be encoded into a single equation. For example, in $\mathcal{FL}_0$ the above system can be encoded

[2]Note, however, that this depends on the expressibility of $\leftrightarrow$ in the logic, i.e., it is no necessarily true for sub-Boolean modal logics.

[3]Turning all concept names into variables would often create too many false positives, i.e., pairs of concept descriptions that unify, but are not meant to define the same intuitive concept.

into the single equation

$$\forall r_1.C_1 \sqcap \ldots \sqcap \forall r_n.C_n \equiv^? \forall r_1.D_1 \sqcap \ldots \sqcap \forall r_n.D_n,$$

where $r_1, \ldots, r_n$ are n distinct role names [13]. In the uni-modal logics considered in Section 3, this approach for encoding a system of equations into a single equation is not possible. However, there we can use the fact that an "equation" is without loss of generality just a formula that we want to make valid. In fact, given formulae $A_1, \ldots, A_1$, we have that σ is a unifier of each of them iff it is a unifier of their conjunction.

3 Unification in modal logic

In this section, we introduce unification from a strictly modal perspective: after summarizing relevant modal systems (just to fix the terminology), we introduce and motivate unification problems; then we start investigating Boolean unification with the aim of extending basic ideas and techniques from Boolean to modal unification. When making such extensions, we shall encounter undecidability limits as well as remarkable positive results and open problems. The connection to admissibility of inference rules will be exploited and considered a relevant source of applications for unification from the symbolic logic point of view.

3.1 Basic modal systems

Recall from Subsection 2.1 that modal propositional formulae are built from $\bot, \top$ and the propositional variables

$$x_1, x_2, \ldots, y_1, y_2, \ldots, z_1, z_2, \ldots$$

by using the connectives $\wedge, \vee, \neg, \Box, \Diamond$.

Syntax

An *axiom system* L is a set of formulae closed under substitution, containing all classical tautologies and all the instances of the Aristotle axiom $\Box(x \to y) \to (\Box x \to \Box y)$. We say that a formula A is *derivable* from the axiom system L (written $\vdash_L A$) iff there is an L-derivation ending in A, where an L-derivation is a list of formulae whose members are either taken from L, or come from previous members of the list by applying an instance of the modus ponens $(x, x \to y / y)$ or of the necessitation $(x / \Box x)$ rules. The (global) *L-entailment* relation $A \Vdash_L B$ means that there is an $L \cup \{A\}$-derivation of B.[4]

[4]That is, B is derived with the help of formulae from L and A by using modus ponens and necessitation (*no* substitution instance of A can be used).

A *(normal, modal) logic* is the set of formulae which are derivable from an axiom system (in the following, we will often not distinguish between an axiom system and the resulting logic). The minimum modal logic is called K; the logics $K4$, GL, $S4$, Grz are obtained from an axiom system for K by adding to it the corresponding axiom schema, i.e., all substitution instances of the respective axioms listed below:

$$K4: \ \Box x \to \Box\Box x \qquad GL: \ \ K4 \wedge (\Box(\Box x \to x) \to \Box x)$$
$$S4: \ K4 \wedge (\Box x \to x) \qquad Grz: \ S4 \wedge (\Box(\Box(x \to \Box x) \to x) \to x)$$

We shall also consider modal logics endowed with a *universal modality* [36], which corresponds to the universal relation in Kripke semantics (see below). The language of these logics contains an additional unary connective $\forall$ (the formula $\exists A$ is defined as $\neg\forall\neg A$). Axiom systems with universal modality must contain the following additional schemata:

$$\forall x \to x; \qquad \forall x \to \forall\forall x; \qquad x \to \forall\exists x; \qquad \forall x \to \Box x.$$

Derivations in systems with the universal modality allow the further inference rule $x/\!\!\forall x$; the definitions of entailment and of a logic are the same as in the case without universal modality. The logics $K_\forall, K4_\forall, GL_\forall, \dots$ are the universal modality counterparts of $K, K4, \dots$ (they are obtained from the minimum axiom system with universal modality $K_\forall$ by adding it the axiom schemata $K4, GL, \dots$).

Semantics

The notions of a frame and of a model have been introduced in Subsection 2.1. Recall also that, for a class $\mathcal{K}$ of frames, the set $L(\mathcal{K})$ of formulae A such that $\mathsf{F} \models A$ holds for all $\mathsf{F} \in \mathcal{K}$ is a modal logic; modal logics of the kind $L(\mathcal{K})$ for some $\mathcal{K}$ are said to be *Kripke complete*. For instance, K is complete for the class of all frames, $K4$ for the class of all transitive frames, $S4$ for the class of all reflexive and transitive frames; the logics GL and Grz are also Kripke complete, namely with respect to all finite transitive irreflexive frames and all finite posets, respectively.

For the universal modality case, we need the further truth-condition (besides those listed in Figure 2)

$$\mathcal{M} \models_w \forall A \quad \text{iff} \quad \text{for all } v, \ \mathcal{M} \models_v A,$$

saying that the relation interpreting the connective $\forall$ is the universal (i.e., the total) relation $W \times W$. Our semantic definitions can be extended in a straightforward way to languages and logics with the universal modality; the above mentioned completeness results also continue to hold.

Unification problems

Fix a logic L (with or without universal modality); following the reasons explained in Subsection 2.2, we define a *unification problem* in L as a formula A and a solution of it as a substitution σ (also called a *unifier for* A) such that $\vdash_L A\sigma$. The set of unifiers of A in the logic L is denoted by $U_L(A)$.

A substitution σ is more general than a substitution τ with respect to a set of variables X and to a logic L (written $\sigma \leq_L^X \tau$) iff there exists a further substitution θ such that for all $x \in X$ we have that $\vdash_L x\tau \leftrightarrow x\sigma\theta$. Similarly, a unifier σ for A is less general than another unifier τ for A iff it holds that $\sigma \leq_L^{Var(A)} \tau$ (where $Var(A)$ is the set of variables occurring in A). The *unification type of a logic* L is now defined in the usual way [17], by referring to the preordered sets $(U_L(A), \leq_L^{Var(A)})$, varying the unification problem (formula) A.

3.2 Motivating unification from the modal logician point of view

The interesting application of unification theory to traditional modal logic concerns recognizability of admissible rules. An inference rule

$$\frac{A}{B} \tag{1}$$

is *admissible* in a logic L iff for every substitution σ,

$$\vdash_L A\sigma \quad \Rightarrow \quad \vdash_L B\sigma.$$

For instance, the Löb rule

$$\frac{\Box x \rightarrow x}{x} \tag{2}$$

is admissible in the system GL; another example is the rule

$$\frac{\Box x}{x} \tag{3}$$

which is admissible in all the systems introduced in Subsection 3.1; a more complicated example is the Lemmon-Scott rule

$$\frac{\Box(\Box(\Box\Diamond\Box x \rightarrow x) \rightarrow (\Box x \vee \Box\neg\Box x))}{\Box\Diamond\Box x \vee \Box\neg\Box x} \tag{4}$$

which is admissible in $S4$ and Grz. Notice that, if $A \Vdash_L B$, then the rule (1) is certainly admissible, but the point is that in non-trivial examples (like the last one) the implication in the opposite direction is not true.

The problem of recognizing admissible rules *effectively* is a standard problem in mathematical logic (already appearing in a classical problems list by Friedman [29]

in the Seventies). Admissible rules can be useful to improve *concrete* deduction, but can also shed light on logical axiomatizations. Take for instance the case of the system GL: it can be shown that the Löb rule can equivalently replace the axiom $\Box(\Box x \to x) \to \Box x$ we introduced in Subsection 3.1 and it is important to have at hand both the original axiom system and the new system. In fact, the original axiom system shows that GL has an *equational* class of modal algebras as an algebraic counterpart, whereas the Löb rule supports an *intuitive understanding* of the logic (interpreting the Löb rule in Kripke semantics terms, one gets a modal formulation of noetherian induction; moreover the Löb rule gives easily Gödel's second incompleteness theorem from completeness of arithmetical interpretation).

Recognizing admissible inferences is a hard task: the problem was solved positively for intuitionistic logic and for basic *transitive* modal systems by V.V. Rybakov (see [58, 60, 61]). Unification theory can provide an alternative and elegant solution to the recognazability of inference rules. Suppose, in fact, that we have proved that a system L has finitary unification type and that we have a type-conformant unification algorithm, i.e., an algorithm that computes a finite complete set of unifiers $U_L(A)$ for the unification problem A. Then it is evident that the rule (1) is admissible iff we have $\vdash_L B\sigma$ for every $\sigma \in U_L(A)$. Clearly, if L is decidable, this is an algorithm for recognizing admissible rules. Thus, unification type *finitarity* turns out to be the key feature here (together, of course, with decidability of L).

There is a second (more subtle) point along which unification theory intersects the mathematics of modal logics and which is related to algebraic aspects: people working in algebra know the importance of *projective* algebras and of *finitely presented algebras* and it has turned out that finitely presented projective algebras are a key ingredient for modal unification. Since we prefer not to introduce here explicitly the algebraic apparatus,[5] we translate below all the needed definitions into symbolic terminology (see also [34]).

3.3 Boolean unification and the universal modality

An interesting feature of modal unification is that it is surprisingly close to Boolean unification: in fact, we shall see that most general and minimal modal unifiers are obtained by Boolean unifiers and their iterations.

First, let us introduce the promised projectivity notion: a formula A is projective (in a logic L) iff there is a unifier σ for A such that

$$A \Vdash_L x \leftrightarrow x\sigma \tag{5}$$

for all variables x occurring in A. Any unifier σ satisfying (5) is called a *projective unifier*. The definition of a projective formula translates into symbolic logic the

[5] this apparatus is, e.g., employed in [30].

statement "the free algebra over $Var(A)$ divided by the congruence generated by the equation $A = \top$ is projective": in fact, since projective algebras can be equivalently defined as retract of free algebras, being a retract means precisely that there is a projective unifier. On the other hand, projective unifiers corresponds to the so-called "reproductive solutions" known from the research on Boolean unification [50] and are quite close to the "transparent unifiers" introduced by Wronski [74].

Projective unifiers are most general unifiers: if τ unifies A, then we obtain

$$\vdash_L x\tau \leftrightarrow x\sigma\tau$$

just by applying τ to (5). This shows that any unifier τ for A is less general than any projective unifier.

Existence of projective unifiers for every unifiable formula is responsible for the fact that unification is unitary in Boolean algebras and in many interesting (fragments of) non-classical propositional logics (see [30, 74, 34]). To construct projective unifiers in classical logic one can proceed as follows. Suppose that L is classical propositional logic, that A is unifiable in L and that γ is a ground unifier for it (notice that, up to Boolean equivalence, the range of γ is $\{\top, \bot\}$, hence γ can be seen as a propositional valuation). Let λ^γ_A be the substitution whose domain is $Var(A)$ and which is defined as

$$x \;\longmapsto\; (A \wedge x) \vee (\neg A \wedge x\gamma). \tag{6}$$

for all $x \in Var(A)$. The following lemma can be shown by a simple induction on the structure of the formula B:

Lemma 1. *Let V be a Boolean evaluation; for every formula B, we have that*

(i) *if $V(A) = \bot$, then $V(B\lambda^\gamma_A) = V(B\gamma)$;*

(ii) *if $V(A) = \top$, then $V(B\lambda^\gamma_A) = V(B)$.*

From this lemma, it follows immediately that, if A is unifiable, then A is projective with projective unifier λ^γ_A (thus, Boolean unification is unitary). In fact, by (ii), we have that $A \Vdash_L x \leftrightarrow x\lambda^\gamma_A$ holds for all $x \in Var(A)$ and by (i) and (ii) we get that $A\lambda^\gamma_A$ is a tautology (it is true under V both in case $V(A) = \bot$ and in case $V(A) = \top$).

The formula (6) is due to Löwenheim: it is precisely the *Löwenheim formula* taken from [50] (we slightly modified it because we do not use Boolean ring notation and because the standard meaning of our unification problems is $A \leftrightarrow^? \top$ rather than $A \leftrightarrow^? \bot$).

The surprising point is that the simple argument of Lemma 1 lifts trivially to the universal modality case. Let in fact L be a logic with universal modality

which is Kripke complete. Without loss of generality, we can now limit ourselves to unification problems of the kind $\forall B$ (this is because for all σ we have that $\vdash_L B\sigma$ iff $\vdash_L (\forall B)\sigma$). Thus let A be a formula whose top-level connective is the universal modality, let γ be a ground unifier of it, and let λ_A^γ be defined as in (6). Then Lemma 1 can be reformulates to

Lemma 2. *Let $\mathcal{M}$ be a Kripke model and let w be a state in it; for every formula B, we have that*

(i) *if $\mathcal{M} \models_w A$, then ($\mathcal{M} \models_w B\lambda_A^\gamma$ iff $\mathcal{M} \models_w B\gamma$);*

(ii) *if $\mathcal{M} \not\models_w A$, then ($\mathcal{M} \models_w B\lambda_A^\gamma$ iff $\mathcal{M} \models_w B$).*

Notice that the proof carries over smoothly because (i) and (ii) do not require mutual induction (the modality is 'universal' hence A is true everywhere or nowhere in $\mathcal{M}$). As a consequence, the equation 'unifiable = projective' (leading to unitarity of unification) can be established for universal modality systems too:

Theorem 3. *Let L be a Kripke complete logic with universal modality; a formula A is unifiable in L iff it is projective in L. As a consequence, unification is unitary in L.*

It should be noticed that the Kripke completeness hypothesis is superfluous, because there is a direct algebraic argument proving Theorem 3 in the general context of discriminator varieties [21].

Can we conclude that we have a method for recognizing admissible rules in any modal logic L with universal modality? Not quite ... we have only reduced admissibility to unifiability,[6] but unifiability itself needs to be investigated. Unfortunately, we are quite far from the goal, as the following negative result shows:

Theorem 4 ([73])**.** *Unifiability of a formula A is undecidable in any logic L with universal modality between $K_\forall$ and $K4_\forall$.*

Notice however that, if $\neg\Box\bot \in L$ (this is the case, for instance, if $L \supseteq S4_\forall$), then there are only finitely many non-equivalent ground formulae in L: in this case, if L is decidable, unifiability can be tested and consequently the Löwenheim formula (6) supplies most general unifiers like in the plain Boolean case.

[6] This reduction was stated and proved in [63], by an argument not too far from the one given above. Notice that the inverse reduction (of unifiability to admissibility) is trivial, because unifiability of A is the same as non-admissibility of the rule $A/\bot$.

3.4 Positive results

In this subsection we consider modal systems L *without* universal modality. The case $L = K$ is *open*, it is the most important open problem in the area. Notice that K cannot have unitary unification, because it is a system enjoying the disjunction property. Let us explain this in more detail. A modal system L has the disjunction property iff for all A, B it holds that

$$\vdash_L \Box A \vee \Box B \quad \Rightarrow \quad \vdash_L A \text{ or } \vdash_L B.$$

If we read this condition from the unification viewpoint, this means in particular that the formula $\Box x \vee \Box \neg x$ has just two (clearly incomparable) unifiers, namely $x \mapsto \top$ and $x \mapsto \bot$.

Positive results for unification have been obtained for transitive logics, by a technique that iterates Löwenheim substitutions. To better understand this technique, let us have a closer look at the formula (6) for Boolean most general unifiers: modulo trivial Boolean manipulations, this formula gives either

$$x \ \longmapsto \ A \wedge x \qquad (\text{in case } x\gamma = \bot) \tag{7}$$

or

$$x \ \longmapsto \ A \to x \qquad (\text{in case } x\gamma = \top). \tag{8}$$

Let us call a substitution a *simplified Löwenheim substitutions* for A if it has domain $Var(A)$ and maps each $x \in Var(A)$ either to $A \wedge x$ or to $A \to x$.[7]

If L contains the transitivity axiom $\Box x \to \Box\Box x$, we can assume without loss of generality that the formulae to be unified have the form $\Box^+ B$, where $\Box^+ B$ is defined as $\Box B \wedge B$: in fact, we have that $\vdash_L B\sigma$ iff $\vdash_L B\sigma \wedge (\Box B)\sigma$ iff $\vdash_L (\Box^+ B)\sigma$ (use the necessitation rule, modus ponens and tautologies). Let A be a *master formula*, i.e. a formula of the kind $\Box^+ B$. It is easy to see that substitutions satisfying condition (5) are closed under composition and that simplified Löwenheim substitutions for A satisfy condition (5); the idea is then to characterize projectivity of A by using *iterations* of simplified Löwenheim substitutions:

Proposition 5 ([32])**.** *Let L be any of the systems $K4, GL, S4, Grz$; a master formula A is projective in L iff a suitable composition of simplified Löwenheim substitutions for A is a unifier for A. There is a primitive recursive bound in the modal degree of A for the required number of such compositions.*

[7] Clearly, if the cardinality of $Var(A)$ is n, then there are 2^n simplified Löwenheim substitutions for A.

We recall that the modal degree $d(C)$ of a formula C counts the number of nested $\Box$ operators in C (formally: $d(x) = x$ for atomic x, $d(\neg C) = d(C)$, $d(C \wedge D) = d(C \vee D) = max(d(C), d(D))$, $d(\Box C) = d(\Diamond C) = d(C) + 1$).

Since the systems $K4, GL, S4, Grz$ are all decidable, Proposition 5 gives a decision procedure for recognizing projectivity and computing projective unifiers. We still need a fact connecting arbitrary unifiers with projective ones:

Theorem 6 ([32]). *Let L be any of the systems $K4, GL, S4, Grz$; any unifier σ of a master formula B is less general than a projective unifier of a projective formula A that L-entails B and has at most the same modal degree as B.*

Notice that, if $A \Vdash_L B$, then any unifier for A is also a unifier for B. Thus we have the following algorithm for computing a minimal complete set of unifiers for a given master formula B: (a) list the (finitely many up to L-equivalence) master formulae A having at most the same modal degree as B; (b) discard those A for which $A \Vdash_L B$ does not hold; (c) determine which of the remaining A are projective by applying the method of Proposition 5. The ($\Vdash_L$-maximal) formulae surviving after step (c) are said to form a *projective* approximation $\Pi(B)$ of B. If $\Pi(B)$ is empty, the formula B is not unifiable; in case it is not empty, the projective unifiers of the formulae in $\Pi(B)$ are a finite complete (minimal) set of unifiers for B. Summing up, we have:

Corollary 7 ([32]). *Let L be any of the systems $K4, GL, S4, Grz$; unification in L is finitary and finite complete sets of unifiers can effectively be computed.*

Note that Proposition 5, Theorem 6, and Corollary 7 can be extended to transitive logics L having finite model property and satisfying a suitable "extensibility" semantic condition [32, 42].

The algorithm sketched above (which is based on Proposition 5 and Theorem 6) has a non-elementary complexity, but it can be greatly simplified: in fact, for most applications, there is no need for computing a finite minimal complete set of unifiers, projective approximations can be computed instead. For instance, in order to recognize that a rule A/B is admissible in a logic L, it is sufficient to check that $C \Vdash_L B$ holds for every $C \in \Pi(A)$.[8] Thus, for instance, the admissibility of the Lemmon-Scott rule (4) (in $S4$ and Grz) comes from the fact that $\{\Box\Diamond\Box x,\ \Box\neg\Box x\}$ is a projective approximation of the premise. Further examples, illustrating in detail various phenomena, can be found in the last section of [32].

The computation of projective approximations and the check of projectivity of a given formula can be performed by semantic methods, without building non-elementarily complex substitutions. To this aim, one uses the following characterization of projectivity in finite Kripke models [32]:

[8] This comes from the general (easy) fact that, for projective C, the condition $C \Vdash_L B$ is equivalent to the fact that a projective unifier of C unifies B.

> A master formula A is projective in L iff it is possible, in any finite Kripke model for L, to make A everywhere true by modifying the evaluation of the atomic formulae only in the part of the model in which A is not true.

Replacing 'master formula' simply by 'formula', it can be shown that the above observation, as well as Proposition 5 and Theorem 6, hold for intuitionistic propositional logic too (see [31]). Based on this observation, a calculus combining tableau- and resolution-steps has been built in [33] for the intuitionistic case: this calculus checks whether a formula is projective in exponential time and builds projective approximations in double exponential time. Many examples (ranging from small to medium size) are worked out in full details in [33]. The extension of this calculus to the modal systems $K4$ and $S4$ has been presented in [75].

For further research along this line, connecting unification and projectivity, see [35], where a natural family of transitive systems enjoying unitary unification is identified. Finally, [28] connects unification types and splittings in the lattice of logics.

3.5 Further recent work on admissible rules

Recent research on admissible inference rules is mostly concerned with the following problems: (i) to find admissibility bases; (ii) to determine the complexity of the recognizability problem; (iii) to build specific sequent-like calculi for admissibility. These research lines often make crucial use of the results and techniques concerning unification and projectivity sketched above. Nevertheless, they constitute a subject of their own, which would require a separated survey. Here, we only give a few pointers to the literature.

There are two basic approaches to the analysis of admissible rules that have been followed in the literature: one is the strategy of [61], which relies on the combinatorics of universal frames of finite rank; the other uses projectivity and extension properties of classes of finite frames (i.e., the kind of apparatus we introduced in Subsection 3.4 above).[9] The latter approach has been used in [39, 42] to investigate admissibility bases for intuitionistic logic, $K4, S4, GL, Grz$ (bases for $S4$ were built also in [62] following the former approach). By definition, an *admissibility basis* for a logic L is a set S of admissible rules such that any other admissible rule r is derivable from S (in the sense that the conclusion of r is L-entailed from the assumption, if L-deduction is enlarged so that it can apply also substitution instances of rules in S, besides axioms from L and standard rules like modus ponens and necessitation); an *independent* admissibility basis is an admissibility basis which is minimal. For the logics $K4, S4, GL, Grz$, for

[9] For the sake of completeness, we mention a third very recent approach [45], making use of canonical rules coming from frame descriptions.

which we have shown positive results regarding unification in Subsection 3.4, *finite* admissibility bases actually do *not* exist [59], but independent *infinite* bases are exhibited in [44], by refining previous work from [42].

Concerning point (ii), [43] correctly notices that the algorithm for admissibility given in [33, 75] is in ExpSpace: in fact, to show that the rule (1) is not admissible, one finds a formula $C \in \Pi(A)$ such that $C \nvdash_L B$ does not hold. For systems like intuitionistic logic, $K4$, $S4$, etc., the entailment test is in PSpace and the algorithm computes each of the candidates C (which are all at most exponentially long) in exponential time, so the overall complexity bound is in ExpSpace. In the light of the results from [43] this is sub-optimal:[10] recognizing admissibility in intuitionistic logic and in logics like $K4$, $S4$, etc. is coNExpTime-complete. However, the optimal algorithm, as it often happens, looks less viable than the algorithm of [33, 75] because it is based not on a calculus but on guessing an exponential-size model. This is the reason why the investigations concerning point (iii) play an important role.

The algorithms for admissibility from [33, 75] mix tableaux and resolution; from a proof-theoretic point of view, it would be cleaner to have a pure tableau or a pure sequent-style calculus. This has been achieved in [41] (improving a previous attempt from [40]), where proof systems for admissibility are presented: in such proof systems, the basic objects are sequent rules, not just ordinary sequents. Another approach is developed in [18]: here methods for synthetizing tableau calculi from [65] are applied to a first-order specification of the class of models used in [61] in order to test rule admissibility.

3.6 Unification in classical modal logics

For the sake of completeness, we report here relevant results from the literature on general Boolean unification, by rephrasing them as results concerning classical (possibly non-normal) modal logics. We follow here the definition of a *classical* modal logic given by Segerberg [70] (strictly speaking, Segerberg considered only modal signatures consisting of a single unary modal operator, but nowadays n-ary multi-modal systems are so common that this restriction does not seem to be justified anymore).

We call a *modal signature* Σ_M a set of function symbols (called modal operators) endowed with corresponding arities; modal formulae are now built up from propositional letters by applying to them the Boolean connectives as well as modal operators from Σ_M. An axiom system L (based on the modal signature Σ_M, which is fixed from now on in this subsection) is now just a set of formulae closed under substitution and containing all classical tautologies; an L-derivation

[10] The algorithm from [61] (based on an ad hoc inspection of an exponential number of Kripke models) is in Π_2^E, hence slightly better from the theoretical complexity viewpoint, but it is sub-optimal too.

can apply (besides axioms from L and modus ponens) also the replacement rules

$$\frac{A_1 \leftrightarrow B_1, \ \ldots, \ A_n \leftrightarrow B_n}{f(A_1, \ldots, A_n) \leftrightarrow f(B_1, \ldots, B_n)}$$

(we have one such rule for every $f \in \Sigma_M$ of arity n).

A *(classical modal) logic* is the set of formulae which are derivable from an axiom system. The minimum classical modal logic is called **E**. Unification problems, unifiers, the instantiation preorder, and the unification types of a logic can be defined for classical modal logics exactly as in the case of normal modal logics.

It is easy to see [9] that equivalence in the minimum classical modal logic **E** is axiomatized by the equational axioms defining Boolean algebras, where the signature of Boolean algebras $(\top, \bot, \wedge, \vee, \neg)$ is, however, extended by the modal parameters from Σ_M. Since the modal parameters do not occur in the axioms defining Boolean algebras, the symbols from Σ_M are so-called *free function symbols*. In unification theory, unification problems that may contain additional free function symbols are called *general* unification problems [17]. Thus, unification in the minimum classical modal logic **E** is the same as general Boolean unification. Whereas elementary Boolean unification and Boolean unification with constants are unitary [50], general Boolean unification is finitary [68]. Algorithms that compute finite complete sets of unifiers for general Boolean unification problems are described in [68, 20]. They are based on aproaches for combining unification algorithms for equational theories over disjoint signatures, where one theory is the theory of Boolean algebras with signature $\top, \bot, \wedge, \vee, \neg$ and the other is the free theory (consisting of the trivial equations $f(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$ for every $f \in \Sigma_M$) with signature Σ_M. Of course, algorithms that compute finite complete sets of unifiers can also be used to decide unifiability, but the complexity of the decision procedure obtained this way need not be optimal. The exact complexity of deciding the solvability of a general Boolean unification problem was determined in [5]: it is PSPACE-complete.[11] This complexity result is based on results from [15] obtained in the context of combining decision procedures for unification in the union of equational theories over disjoint signatures.

4 Unification in description logic

The main motivation for introducing unification in description logics was to avoid redundancies in knowledge bases that are built by several knowledge engineers over a long time period. In this setting, it frequently happens that the same (intuitive) concept is introduced several times, often with slightly differing descriptions. Testing for equivalence of concepts is not always sufficient to find out

[11] In contrast, this problem is NP-complete for elementary Boolean unification problems and Π_2^p-complete for Boolean unification problems with constants.

whether, for a given concept description, there already exists another concept description in the knowledge base describing the same notion. As an example, lets us ask whether the following two $\mathcal{ALC}$ concept descriptions might denote the same (intuitive) concept:

$$\forall \mathsf{child}.\forall \mathsf{child}.\mathsf{Rich} \sqcap \forall \mathsf{child}.\mathsf{Rmr} \quad \text{and} \quad \mathsf{Acr} \sqcap \forall \mathsf{child}.\mathsf{Acr} \sqcap \forall \mathsf{child}.\forall \mathsf{spouse}.\mathsf{Rich}.$$

The answer is yes, since replacing the concept name Rmr by the description $\mathsf{Rich} \sqcap \forall \mathsf{spouse}.\mathsf{Rich}$ and Acr by $\forall \mathsf{child}.\mathsf{Rich}$ yields the descriptions

$$\forall \mathsf{child}.\forall \mathsf{child}.\mathsf{Rich} \sqcap \forall \mathsf{child}.(\mathsf{Rich} \sqcap \forall \mathsf{spouse}.\mathsf{Rich}),$$
$$\forall \mathsf{child}.\mathsf{Rich} \sqcap \forall \mathsf{child}.\forall \mathsf{child}.\mathsf{Rich} \sqcap \forall \mathsf{child}.\forall \mathsf{spouse}.\mathsf{Rich},$$

which are obviously equivalent. Thus, under the assumption that Rmr stands for "Rich and married rich" and Acr for "All children are rich", we can conclude that both descriptions are meant to express the concept "All grandchildren are rich and all children are rich and married rich". This connection between the two descriptions can be found by a unification algorithm if we declare Rmr and Acr to be variables.

Assume that $\mathcal{L}$ is some description logic. In order to define unification of concept descriptions, we first introduce the notion of a substitution. To this purpose, we partition the set of concepts names into a set N_v of concept variables (which may be replaced by substitutions) and a set N_c of concept constants (which must not be replaced by substitutions). Intuitively, N_v are the concept names that have possibly been given another name or been specified in more detail in another concept description describing the same notion. The elements of N_c are the ones of which it is assumed that the same name is used by all knowledge engineers (e.g., standardised names in a certain domain).

A *substitution* σ is a mapping from N_v into the set of all $\mathcal{L}$ concept descriptions. Given a concept variable $X \in N_v$, we denote its image under σ by $X\sigma$. This mapping is extended to concept descriptions in the obvious way. For example, in the case of $\mathcal{ALC}$ we have

- $A\sigma := A$ for all $A \in N_c$,
- $\top\sigma := \top$ and $\bot\sigma = \bot$,
- $(C \sqcap D)\sigma := C\sigma \sqcap D\sigma$, $(C \sqcup D)\sigma := C\sigma \sqcup D\sigma$, and
- $(\exists r.C)\sigma := \exists r.(C\sigma)$ and $(\forall r.C)\sigma := \forall r.(C\sigma)$.

An *$\mathcal{L}$-unification problem* is of the form $\Gamma = \{C_1 \equiv^? D_1, \ldots, C_n \equiv^? D_n\}$, where $C_1, D_1, \ldots C_n, D_n$ are $\mathcal{L}$ concept descriptions. The substitution σ is a *unifier* (or *solution*) of Γ iff $C_i\sigma \equiv D_i\sigma$ for $i = 1, \ldots, n$. In this case, Γ is called *solvable* or *unifiable*. When we say that $\mathcal{L}$-unification is *decidable (NP-complete,*

ExpTime-complete), then we mean that the following decision problem is decidable (NP-complete, ExpTime-complete): given an $\mathcal{EL}$-unification problem Γ, decide whether Γ is solvable or not.

In the following, we consider the complexity of this decision problem for the description logics $\mathcal{FL}_0$ and $\mathcal{EL}$.

4.1 Unification in $\mathcal{FL}_0$

Concept descriptions in $\mathcal{FL}_0$ are formed according to the following syntax rule:

$$C, D \longrightarrow A \mid \top \mid C \sqcap D \mid \forall r.C$$

where A ranges over concept names and r ranges over role names. The semantics of the concept constructors used in $\mathcal{FL}_0$ is the one given in Figure 1 for the $\mathcal{ALC}$ constructors.

First, we sketch a structural algorithm that can be used to decide equivalence of $\mathcal{FL}_0$ concept descriptions in polynomial time. This algorithm normalizes the descriptions to be tested for equivalence, and then compares the syntactic structure of the normal forms.

By using the equivalence $\forall r.(C \sqcap D) \equiv \forall r.C \sqcap \forall r.D$ as a rewrite rule from left to right, any $\mathcal{FL}_0$ concept description can be transformed into an equivalent description that is a conjunction[12] of descriptions of the form $\forall r_1.\cdots\forall r_m.A$ for $m \geq 0$ (not necessarily distinct) role names $r_1, \ldots, r_m$ and a concept name A. We abbreviate $\forall r_1.\cdots\forall r_m.A$ by $\forall r_1 \ldots r_m.A$, where $r_1 \ldots r_m$ is viewed as a word over the alphabet of all role names. In addition, instead of $\forall w_1.A \sqcap \ldots \sqcap \forall w_\ell.A$ we write $\forall L.A$ where $L := \{w_1, \ldots, w_\ell\}$ is a finite set of words over the alphabet of all role names. The expression $\forall \emptyset.A$ is considered to be equivalent to the top concept $\top$, which means that it can be added to a conjunction without changing the meaning of the concept. Using these abbreviations, any pair of $\mathcal{FL}_0$ concept descriptions C, D containing the concept names $A_1, \ldots, A_k$ can be rewritten as

$$C \equiv \forall U_1.A_1 \sqcap \ldots \sqcap \forall U_k.A_k \quad \text{and} \quad D \equiv \forall V_1.A_1 \sqcap \ldots \sqcap \forall V_k.A_k,$$

where U_i, V_i are finite sets of words over the alphabet of all role names. This normal form provides us with the following *characterization of equivalence* of $\mathcal{FL}_0$ concept descriptions [13]:

$$C \equiv D \quad \text{iff} \quad U_i = V_i \ \text{ for all } i, 1 \leq i \leq k.$$

Since the size of the normal forms is polynomial in the size of the original descriptions, and since the equality tests $U_i = V_i$ can also be realized in polynomial time, this yields a polynomial-time decision procedure for equivalence in $\mathcal{FL}_0$.

[12] The empty conjunction is $\top$.

The unification procedure for $\mathcal{FL}_0$ developed in [12, 13] crucially depends on the characterization of equivalence introduced above. As mentioned in Section 2, we can without loss of generality restrict the attention to unification problems consisting of a single equation $C \equiv^? D$. Using the normal form of $\mathcal{FL}_0$ concept descriptions introduced above, we can write the descriptions C, D in the form

$$C \equiv \forall S_{0,1}.A_1 \sqcap \ldots \sqcap \forall S_{0,k}.A_k \sqcap \forall S_1.X_1 \sqcap \ldots \sqcap \forall S_n.X_n,$$
$$D \equiv \forall T_{0,1}.A_1 \sqcap \ldots \sqcap \forall T_{0,k}.A_k \sqcap \forall T_1.X_1 \sqcap \ldots \sqcap \forall T_n.X_n,$$

where $A_1, \ldots, A_k$ are the concept constants and $X_1, \ldots, X_n$ the concept variables occurring in C, D, and $S_{0,i}, S_j, T_{0,i}, T_j$ $(i = 1, \ldots, k, j = 1, \ldots, n)$ are finite sets of words over the alphabet of all role names. In [13], it is shown that $C \equiv^? D$ has a unifier iff for all $i = 1, \ldots, k$, the *linear language equation*

$$S_{0,i} \cup S_1 X_{1,i} \cup \cdots \cup S_n X_{n,i} = T_{0,i} \cup T_1 X_{1,i} \cup \cdots \cup T_n X_{n,i}$$

has a solution, i.e., we can substitute the variables $X_{j,i}$ by finite languages such that the equation holds. Note that this is not a system of k equations that must be solved simultaneously: since they do not share variables, each of these equations can be solved separately.

Let us illustrate the connection between $\mathcal{FL}_0$ unification problems and linear language equations by a simple example. The normal forms of the concept descriptions

$$C := \forall r.(A_1 \sqcap \forall r.A_2) \sqcap \forall r.\forall s.X_1 \text{ and } D := \forall r.\forall s.(\forall s.A_1 \sqcap \forall r.A_2) \sqcap \forall r.X_1 \sqcap \forall r.\forall r.A_2$$

are

$$C \equiv \forall\{r\}.A_1 \sqcap \forall\{rr\}.A_2 \sqcap \forall\{rs\}.X_1 \text{ and } D \equiv \forall\{rss\}.A_1 \sqcap \forall\{rsr, rr\}.A_2 \sqcap \forall\{r\}.X_1.$$

Thus, the unification problem $C \equiv^? D$ leads to the two linear language equations

$$\{r\} \cup \{rs\}X_{1,1} = \{rss\} \cup \{r\}X_{1,1},$$
$$\{rr\} \cup \{rs\}X_{1,2} = \{rsr, rr\} \cup \{r\}X_{1,2}.$$

The first equation (the one for A_1) has $X_{1,1} = \{\varepsilon, s\}$ as a solution, and the second (the one for A_2) has $X_{1,2} = \{r\}$ as a solution. These two solutions yield the following unifier of $C \equiv^? D$:

$$\{X_1 \mapsto A_1 \sqcap \forall s.A_1 \sqcap \forall r.A_2\}.$$

By an exponential time reduction to the emptiness problem of top-down automata on finite trees it is shown in [13] that solvability of linear language equations of the form introduced above can be decided in exponential time. ExpTime-hardness is shown by a reduction from the intersection emptiness problem for deterministic top-down tree automata.

Theorem 8 ([13]). *Unification in $\mathcal{FL}_0$ is ExpTime-complete.*

In [10], it is shown that unification in $\mathcal{FL}_{reg}$, which extends $\mathcal{FL}_0$ by the role constructors union, composition, and reflexive-transitive closure, is also ExpTime-complete. Basically, instead of linear language equations over finite sets of words, one obtains linear language equations over regular sets of words, and uses automata working on infinite trees to solve them.

4.2 Unification in $\mathcal{EL}$

Concept descriptions in $\mathcal{EL}$ are formed according to the following syntax rule:

$$C, D \longrightarrow A \mid \top \mid C \sqcap D \mid \exists r.C$$

where A ranges over concept names and r ranges over role names. The semantics of the concept constructors used in $\mathcal{EL}$ is the one given in Figure 1 for the $\mathcal{ALC}$ constructors.

In order to *characterize equivalence of $\mathcal{EL}$ concept descriptions*, the notion of a reduced $\mathcal{EL}$ concept description is introduced in [49]. A given $\mathcal{EL}$ concept description can be transformed into an equivalent reduced description by applying the following rules modulo associativity and commutativity of conjunction:

$C \sqcap \top \to C$	for all $\mathcal{EL}$ concept descriptions C
$A \sqcap A \to A$	for all concept names A
$\exists r.C \sqcap \exists r.D \to \exists r.C$	for all $\mathcal{EL}$ concept descriptions C, D with $C \sqsubseteq D$

Obviously, these rules are equivalence preserving. We say that the $\mathcal{EL}$ concept description C is *reduced* if none of the above rules is applicable to it (modulo associativity and commutativity of $\sqcap$). The $\mathcal{EL}$ concept description D is a *reduced form* of C if D is reduced and can be obtained from C by applying the above rules (modulo associativity and commutativity of $\sqcap$).

Theorem 9. *Let C, D be $\mathcal{EL}$ concept decriptions, and $\widehat{C}, \widehat{D}$ reduced forms of C, D, respectively. Then $C \equiv D$ iff $\widehat{C}$ is identical to $\widehat{D}$ up to associativity and commutativity of $\sqcap$.*

Since subsumption in $\mathcal{EL}$ can be decided in polynomial time [6, 7], equivalence of $\mathcal{EL}$ concept decriptions can also be decided in polynomial time, and the reduced form of a given $\mathcal{EL}$ concept decription can be computed in polynomial time.

The proof that unification in $\mathcal{EL}$ is an NP-complete problem [11] proceeds in several steps:

1. a normal form for $\mathcal{EL}$ unification problems (the so-called flat form) as well as the notion of an atom occurring in such a flat unification problem are introduced;

2. a well-founded ordering on ground substitutions is introduced, and it is shown that every solvable $\mathcal{EL}$ unification problem has a ground unifier that is minimal w.r.t. this ordering;

3. it is shown that minimal ground unifiers are local in the sense that they introduce only atoms occurring in the input $\mathcal{EL}$ unification problems in flat form;

4. to decide unifiability, one thus guesses a ground substitution that is local in this sense, and then tests whether it is a unifier;

5. since the substitutions generated in the previous step may be of exponential size, one actually needs to employ structure sharing to obtain an algorithm that runs in non-deterministic polynomial time.

Let us now look at these steps in a bit more detail. An $\mathcal{EL}$ concept description is called an *atom* iff it is a concept name (i.e., concept constant or concept variable) or an existential restriction $\exists r.D$. Obviously, any $\mathcal{EL}$ concept description is (equivalent to) a conjunction of atoms, where the empty conjunction is $\top$. The set $At(C)$ of *atoms of an $\mathcal{EL}$ concept description* C is defined inductively: if $C = \top$, then $At(C) := \emptyset$; if C is a concept name, then $At(C) := \{C\}$; if $C = \exists r.D$ then $At(C) := \{C\} \cup At(D)$; if $C = C_1 \sqcap C_2$, then $At(C) := At(C_1) \cup At(C_2)$. Concept names and existential restrictions $\exists r.D$ where D is a concept name or $\top$ are called *flat atoms*. The $\mathcal{EL}$-unification problem Γ is *flat* iff it only contains equations of the following form:

- $X \equiv^? C$ where X is a variable and C is a non-variable flat atom;
- $X_1 \sqcap \ldots \sqcap X_m \equiv^? Y_1 \sqcap \ldots \sqcap Y_n$ where $X_1, \ldots, X_m, Y_1, \ldots, Y_n$ are variables.

By introducing new concept variables and eliminating $\top$, any $\mathcal{EL}$-unification problem Γ can be transformed in polynomial time into a flat $\mathcal{EL}$-unification problem Γ' such that Γ is solvable iff Γ' is solvable. Thus, we may assume without loss of generality that our input $\mathcal{EL}$-unification problems are flat. Given a flat $\mathcal{EL}$-unification problem $\Gamma = \{C_1 \equiv^? D_1, \ldots, C_n \equiv^? D_n\}$, we call the atoms of $C_1, D_1, \ldots, C_n, D_n$ the *atoms of* Γ.

The unifier σ of Γ is called *reduced* (*ground*) iff, for all concept variables X occurring in Γ, the $\mathcal{EL}$-concept description $X\sigma$ is reduced (does not contain variables). Obviously, Γ is solvable iff it has a reduced ground unifier.

Next, we define a well-founded ordering on ground unifiers by comparing the concept description in their ranges w.r.t. the *inverse subsumption order*: given $\mathcal{EL}$-concept descriptions C, D, we define $C >_{is} D$ iff $C \sqsubset D$. In [11] it is shown that the strict order $>_{is}$ defined this way is well-founded. Consequently, its multiset extension $>_m$ is also well-founded [14]. Given a ground unifier σ of Γ, we consider

the multiset $S(\sigma)$ of all $\mathcal{EL}$-concept descriptions $X\sigma$, where X ranges over all concept variables occurring in Γ. For two ground unifiers σ, θ of Γ, we define $\sigma \succ \theta$ iff $S(\sigma) >_m S(\theta)$. The ground unifier σ of Γ is *minimal* iff there is no ground unifier θ of Γ such that $\sigma \succ \theta$.

As an easy consequence of the fact that $\succ$ is well-founded we obtain that an $\mathcal{EL}$-unification problem Γ is solvable iff it has a minimal reduced ground unifier. The following proposition shows that minimal reduced ground unifiers of flat $\mathcal{EL}$-unification problems satisfy a locality property that makes it easy to check (with an NP-algorithm) whether such a unifier exists or not.

Proposition 10. *Let Γ be a flat $\mathcal{EL}$-unification problem and γ a minimal reduced ground unifier of Γ. If X is a concept variable occurring in Γ, then $\gamma(X) \equiv \top$ or there are non-variable atoms $D_1, \ldots, D_n$ ($n \geq 1$) of Γ such that $\gamma(X) \equiv \gamma(D_1) \sqcap \ldots \sqcap \gamma(D_n)$.*

This proposition suggests the following *non-deterministic algorithm for deciding solvability of a given flat $\mathcal{EL}$-unification problem Γ:*

1. For every variable X occurring in Γ, guess a finite, possibly empty, set S_X of non-variable atoms of Γ.

2. We say that the variable X *directly depends on* the variable Y if Y occurs in an atom of S_X. Let *depends on* be the transitive closure of *directly depends on*. If there is a variable that depends on itself, then the algorithm returns "fail." Otherwise, there exists a strict linear order $>$ on the variables occurring in Γ such that $X > Y$ if X depends on Y.

3. We define the substitution σ along the linear order $>$:

   - If X is the least variable w.r.t. $>$, then S_X does not contain any variables. We define $X\sigma$ to be the conjunction of the elements of S_X, where the empty conjunction is $\top$.
   - Assume that $Y\sigma$ is defined for all variables $Y < X$. Then S_X contains only variables Y for which $Y\sigma$ is already defined. If S_X is empty, then we define $X\sigma := \top$. Otherwise, let $S_X = \{D_1, \ldots, D_n\}$. We define $X\sigma := D_1\sigma \sqcap \ldots \sqcap D_n\sigma$.

4. Test whether the substitution σ computed in the previous step is a unifier of Γ. If this is the case, then return σ; otherwise, return "fail."

This algorithm is trivially *sound* since it only returns substitutions that are unifiers of Γ. In addition, it obviously always terminates. In [11], completeness is shown by proving that, for every solvable flat $\mathcal{EL}$ unification problem Γ, there is a way of guessing in Step 1 subsets S_X of the non-variable atoms of Γ such that the *depends on* relation determined in Step 2 is acyclic and the substitution σ computed in Step 3 is a unifier of Γ.

Theorem 11. *$\mathcal{EL}$-unification is NP-complete.*

NP-hardness follows from the fact that already $\mathcal{EL}$-matching (where only one side of the equation contains variables) is NP-complete [49]. To show that the problem can be decided by a non-deterministic polynomial-time algorithm, we analyse the complexity of the algorithm sketched above. Obviously, guessing the sets S_X (Step 1) can be done within NP. Computing the *depends on* relation and checking it for acyclicity (Step 2) is clearly polynomial.

Steps 3 and 4 are more problematic. In fact, since a variable may occur in different atoms of Γ, the substitution σ computed in Step 3 may be of exponential size. This is actually the same reason that makes a naïve algorithm for syntactic unification compute an exponentially large most general unifier [17]. As in the case of syntactic unification, the solution to this problem is basically structure sharing, i.e., in (a representation of) the substitution σ, identical subdescriptions are shared. The fact that equivalences $C\sigma \equiv D\sigma$ can be checked in time polynomial in the size of such a compact representation of σ is an easy consequence of the results in [6], where it is shown that subsumption (and thus equivalence) in $\mathcal{EL}$ w.r.t. so-called acyclic TBoxes can be decided in polynomial time (see [11] for more details).

5 Conclusion

Unification in modal logic is a well-established research area that already produced interesting deep results. Still, besides solving the big open problem concerning unification in K (where basically nothing is known), further work is needed especially at the syntactic level, where satisfactory calculi for unification and for admissibility must be designed and implemented, following the most recent research trends mentioned in Subsection 3.5. In the end, it would be desirable to embed such calculi into decision procedures and reasoning tools for modal logics, in order for unification to gain a rôle similar to the one it already plays in classical automated reasoning.

Compared to the host of results on unification in modal logics, the investigation of unification in description logics is only at its beginning. Currently, the only positive results are the ones for the two inexpressive DLs $\mathcal{EL}$ and $\mathcal{FL}_0$. Whereas $\mathcal{FL}_0$ does not appear to be relevant for applications since there are almost no terminologies (nowadays usually called ontologies) formulated in $\mathcal{FL}_0$, the situation is quite different for $\mathcal{EL}$. For example, both the large medical ontology SNOMED CT[13] and the Gene Ontology[14] can be expressed in $\mathcal{EL}$, and the same is true for large parts of the medical ontology GALEN [56]. The importance of $\mathcal{EL}$ can

[13] http://www.ihtsdo.org/snomed-ct/

[14] http://www.geneontology.org/

also be seen from the fact that the new OWL 2 standard[15] contains a sub-profile OWL 2 EL, which is based on (an extension of) $\mathcal{EL}$. In the context of medical ontologies, the redundancy problem that we have mentioned as a motivation for considering unification in description logics actually occurs. For example, in [22], two different extensions of SNOMED CT by so-called post-coordinated concepts were considered. The authors used an (incomplete) equivalence test to find out how large the overlap between the two extensions is (i.e., how many of the new concepts belonged to both extensions). As pointed out in the introduction, the equivalence test cannot deal with situations where different knowledge engineers use different names for concepts, or model on different levels of granularity. It is thus interesting to find out whether using unifiability rather than equivalence yields a larger overlap.

Of course, any result for unification in modal logics can be viewed as a result for unification in a corresponding description logic. However, the description logics that correspond to the modal logics for which there are positive results for unification, like $K4, GL, S4, Grz$, do not appear to be of great relevance in applications of description logics. In contrast, the undecidability result for unification in modal logics with universal modality between $K_\forall$ and $K4_\forall$ [73] also implies undecidability of unification in some expressive and application-relevant DLs (e.g., the DL $\mathcal{SHIQ}$ [38]). As mentioned above, the big open problem for unification in modal logics is decidability of unification in the modal logic K. Since the multi-modal variant of K corresponds to the application-relevant description logic $\mathcal{ALC}$, a positive result for K would also have a big impact on unification in description logics and its applications.

References

[1] S. Anantharaman, P. Narendran, and M. Rusinowitch. Intruders with caps. In F. Baader, editor, *Proceedings of the 18th International Conference on Rewriting Techniques and Applications (RTA 2007)*, volume 4533 of *Lecture Notes in Computer Science*, pages 20–35. Springer-Verlag, 2007.

[2] Y. Auffray and P. Enjalbert. Modal theorem proving: An equational viewpoint. *J. Logic and Computation*, 2(3):247–295, 1992.

[3] F. Baader. Unification in idempotent semigroups is of type zero. *J. Automated Reasoning*, 2(3):283–286, 1986.

[4] F. Baader. Unification in commutative theories. *J. Symbolic Computation*, 8(5):479–497, 1989.

[15] See http://www.w3.org/TR/owl2-profiles/

[5] F. Baader. On the complexity of Boolean unification. *Information Processing Letters*, 67(4):215–220, 1998.

[6] F. Baader. Terminological cycles in a description logic with existential restrictions. In G. Gottlob and T. Walsh, editors, *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003)*, pages 325–330, Acapulco, Mexico, 2003. Morgan Kaufmann, Los Altos.

[7] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In L. P. Kaelbling and A. Saffiotti, editors, *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, pages 364–369, Edinburgh (UK), 2005. Morgan Kaufmann, Los Altos.

[8] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[9] F. Baader, S. Ghilardi, and C. Tinelli. A new combination procedure for the word problem that generalizes fusion decidability results in modal logics. *Information and Computation*, 204(10):1413–1452, 2006.

[10] F. Baader and R. Küsters. Unification in a description logic with transitive closure of roles. In R. Nieuwenhuis and A. Voronkov, editors, *Proc. of the 8th Int. Conf. on Logic for Programming and Automated Reasoning (LPAR 2001)*, volume 2250 of *Lecture Notes in Artificial Intelligence*, pages 217–232, Havana, Cuba, 2001. Springer-Verlag.

[11] F. Baader and B. Morawska. Unification in the description logic $\mathcal{EL}$. In R. Treinen, editor, *Proceedings of the 20th International Conference on Rewriting Techniques and Applications (RTA 2009)*, volume 5595 of *Lecture Notes in Computer Science*, pages 350–364. Springer-Verlag, 2009.

[12] F. Baader and P. Narendran. Unification of concept terms in description logics. In H. Prade, editor, *Proc. of the 13th Eur. Conf. on Artificial Intelligence (ECAI'98)*, pages 331–335. John Wiley & Sons, 1998.

[13] F. Baader and P. Narendran. Unification of concepts terms in description logics. *J. of Symbolic Computation*, 31(3):277–305, 2001.

[14] F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, United Kingdom, 1998.

[15] F. Baader and K. U. Schulz. Unification in the union of disjoint equational theories: Combining decision procedures. *J. Symbolic Computation*, 21:211–243, 1996.

[16] F. Baader and J. H. Siekmann. Unification theory. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, pages 41–125. Oxford University Press, Oxford, UK, 1994.

[17] F. Baader and W. Snyder. Unification theory. In J. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, pages 447–533. Elsevier Science Publishers, 2001.

[18] S. Babenyshev, R. Vladimir, R. Schmidt, and D. Tishkovsky. A tableau method for checking rule admissibility in $S4$. In *Proc. of the 6th Workshop on Methods for Modalities (M4M-6)*, Copenhagen, 2009.

[19] P. Blackburn, J. van Benthem, and F. Wolter, editors. *The Handbook of Modal Logic*. Elsevier, 2006.

[20] A. Boudet, J.-P. Jouannaud, and M. Schmidt-Schauß. Unification in Boolean rings and Abelian groups. *J. Symbolic Computation*, 8:449–477, 1989.

[21] S. Burris. Discriminator varieties and symbolic computation. *J. of Symbolic Computation*, 13(2):175–207, 1992.

[22] J. R. Campbell, A. Lopez Osornio, F. de Quiros, D. Luna, and G. Reynoso. Semantic interoperability and SNOMED CT: A case study in clinical problem lists. In K. Kuhn, J. Warren, and T.-Y. Leong, editors, *Proc. of the 12th World Congress on Health (Medical) Informatics (MEDINFO 2007)*, pages 2401–2402. IOS Press, 2007.

[23] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP decision procedure for protocol insecurity with XOR. In *Proceedings of the 18th IEEE Symposium on Logic in Computer Science (LICS 2003)*, pages 261–270, 2003.

[24] H. Comon-Lundh and V. Shmatikov. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In *Proceedings of the 18th IEEE Symposium on Logic in Computer Science (LICS 2003)*, pages 271–280, 2003.

[25] V. Cortier, S. Delaune, and P. Lafourcade. A survey of algebraic properties used in cryptographic protocols. *Journal of Computer Security*, 14(1):1–43, 2006.

[26] L. F. del Cerro and A. Herzig. Automated quantified logic. In P. Brazdil and K. Konolige, editors, *Machine Learning, Meta-Reasoning and Logics*, pages 301–317. Kluwer, 1989.

[27] S. Delaune. Easy intruder deduction problems with homomorphisms. *Inf. Process. Lett.*, 97(6):213–218, 2006.

[28] W. Dzik. Splittings of lattices of theories and unification types. In *Contributions to general algebra. 17*, pages 71–81. Heyn, Klagenfurt, 2006.

[29] H. Friedman. One hundred and two problems in mathematical logic. *J. Symbolic Logic*, 40:113–129, 1975.

[30] S. Ghilardi. Unification through projectivity. *J. Logic Comput.*, 7(6):733–752, 1997.

[31] S. Ghilardi. Unification in intuitionistic logic. *J. Symbolic Logic*, 64(2):859–880, 1999.

[32] S. Ghilardi. Best solving modal equations. *Ann. Pure Appl. Logic*, 102(3):183–198, 2000.

[33] S. Ghilardi. A resolution/tableaux algorithm for projective approximations in IPC. *Log. J. IGPL*, 10(3):229–243, 2002.

[34] S. Ghilardi. Unification, finite duality and projectivity in varieties of Heyting algebras. *Ann. Pure Appl. Logic*, 127(1-3):99–115, 2004. Provinces of logic determined.

[35] S. Ghilardi and L. Sacchetti. Filtering unification and most general unifiers in modal logic. *J. Symbolic Logic*, 69(3):879–906, 2004.

[36] V. Goranko and S. Passy. Using the universal modality: gains and questions. *J. Logic Comput.*, 2(1):5–30, 1992.

[37] A. Herzig. *Raisonnement Automatique en Logique Modale et Algorithmes D'Unification*. Ph.d. thesis, Univ. Paul-Sabatier, Toulouse, 1989.

[38] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for very expressive description logics. *J. of the Interest Group in Pure and Applied Logic*, 8(3):239–264, 2000.

[39] R. Iemhoff. On the admissible rules of intuitionistic propositional logic. *J. Symbolic Logic*, 66(1):281–294, 2001.

[40] R. Iemhoff. Towards a proof system for admissibility. In *Computer science logic*, volume 2803 of *Lecture Notes in Comput. Sci.*, pages 255–270. Springer, Berlin, 2003.

[41] R. Iemhoff and G. Metcalfe. Proof theory for admissible rules. *Ann. Pure Appl. Logic*, 159(1-2):171–186, 2009.

[42] E. Jeřábek. Admissible rules of modal logics. *J. Logic Comput.*, 15(4):411–431, 2005.

[43] E. Jeřábek. Complexity of admissible rules. *Arch. Math. Logic*, 46(2):73–92, 2007.

[44] E. Jeřábek. Independent bases of admissible rules. *Log. J. IGPL*, 16(3):249–267, 2008.

[45] E. Jeřábek. Canonical rules. *J. Symbolic Logic*, 2009. (to appear).

[46] J.-P. Jouannaud and C. Kirchner. Solving equations in abstract algebras: A rule-based survey of unification. In J.-L. Lassez and G. Plotkin, editors, *Computational Logic: Essays in Honor of A. Robinson*. MIT Press, Cambridge, MA, 1991.

[47] D. Kapur, P. Narendran, and L. Wang. An E-unification algorithm for analyzing protocols that use modular exponentiation. In R. Nieuwenhuis, editor, *Proceedings of the 14th International Conference on Rewriting Techniques and Applications (RTA 2003)*, volume 2706 of *Lecture Notes in Computer Science*, pages 165–179. Springer-Verlag, 2003.

[48] D. E. Knuth and P. B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*. Pergamon Press, Oxford, 1970.

[49] R. Küsters. *Non-standard Inferences in Description Logics*, volume 2100 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2001.

[50] U. Martin and T. Nipkow. Boolean unification—the story so far. In *Unification*, pages 437–455. Academic Press, London, 1990.

[51] H. J. Ohlbach. A resolution calculus for modal logics. In E. Lusk and R. Overbeek, editors, *Proceedings of the 9th International Conference on Automated Deduction*, volume 310 of *Lecture Notes in Computer Science*, pages 500–516, Argonne, IL, 1988. Springer-Verlag.

[52] H. J. Ohlbach. Semantics-based translation methods for modal logics. *J. Logic and Computation*, 1(5):691–746, 1991.

[53] H. J. Ohlbach and R. A. Schmidt. Functional translation and second-order frame properties of modal logics. *J. Logic and Computation*, 7(5):581–603, 1997.

[54] G. Peterson and M. E. Stickel. Complete sets of reductions for equational theories with complete unification algorithms. *J. of the ACM*, 28(2):233–264, 1981.

[55] G. Plotkin. Building in equational theories. *Machine Intelligence*, 7:73–90, 1972.

[56] A. Rector and I. Horrocks. Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In *Proceedings of the Workshop on Ontological Engineering, AAAI Spring Symposium (AAAI'97)*, Stanford, CA, 1997. AAAI Press.

[57] J. A. Robinson. A machine oriented logic based on the resolution principle. *J. of the ACM*, 12(1):23–41, 1965.

[58] V. V. Rybakov. A criterion for admissibility of rules in the modal system S4 and intuitionistic logic. *Algebra and Logic*, 23(5):369–384, 1984. (English translation).

[59] V. V. Rybakov. The bases for admissible rules of logics S4 and Int. *Algebra and Logic*, 24:55–68, 1985. (English translation).

[60] V. V. Rybakov. Rules of inference with parameters for intuitionistic logic. *J. Symbolic Logic*, 57(3):912–923, 1992.

[61] V. V. Rybakov. *Admissibility of logical inference rules*, volume 136 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 1997.

[62] V. V. Rybakov. Construction of an explicit basis for rules admissible in modal system S4. *MLQ Math. Log. Q.*, 47(4):441–446, 2001.

[63] V. V. Rybakov. Multi-modal and temporal logics with universal formula—reduction of admissibility to validity and unification. *J. Logic Comput.*, 18(4):509–519, 2008.

[64] K. Schild. A correspondence theory for terminological logics: Preliminary report. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91)*, pages 466–471, 1991.

[65] R. Schmidt and D. Tishkovsky. Automated synthesis of tableau calculi. In M. Giese and A. Waaler, editors, *Proceedings of the 18th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (Tableaux 2009)*, volume 5607 of *Lecture Notes in Computer Science*, pages 438–451, Oslo, Norway, 2009. Springer-Verlag.

[66] R. A. Schmidt. E-unification for subsystems of $S4$. In T. Nipkow, editor, *Proceedings of the 9th International Conference on Rewriting Techniques and Applications*, volume 1379 of *Lecture Notes in Computer Science*, pages 106–120, Tsukuba, Japan, 1998. Springer-Verlag.

[67] M. Schmidt-Schauß. Unification under associativity and idempotence is of type nullary. *J. Automated Reasoning*, 2(3):277–282, 1986.

[68] M. Schmidt-Schauß. Unification in a combination of arbitrary disjoint equational theories. *J. Symbolic Computation*, 8(1,2):51–99, 1989.

[69] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[70] K. Segerberg. *An Essay in Classical Modal Logic*, volume 13 of *Filosofiska Studier*. Uppsala Universitet, 1971.

[71] J. H. Siekmann. Unification theory: A survey. *J. Symbolic Computation*, 7(3,4):207–274, 1989.

[72] V. Sofronie-Stokkermans. Locality and subsumption testing in $\mathcal{EL}$ and some of its extensions. In *Proc. Advances in Modal Logic (AiML'08)*, 2008.

[73] F. Wolter and M. Zakharyaschev. Undecidability of the unification and admissibility problems for modal and description logics. *ACM Trans. Comput. Log.*, 9(4):Art. 25, 20, 2008.

[74] A. Wroński. Transparent unification problem. *Rep. Math. Logic*, 29:105–107, 1995. First German-Polish Workshop on Logic & Logical Philosophy (Bachotek, 1995).

[75] D. Zucchelli. Studio e realizzazione di algoritmi per l'unificazione nelle logiche modali. Master's thesis in computer science, Università degli Studi di Milano, 2004.