First-Order Resolution Methods for Modal Logics[⋆]

R. A. Schmidt[1] and U. Hustadt[2]

[1] The University of Manchester, UK, Renate.Schmidt@manchester.ac.uk
[2] University of Liverpool, UK, U.Hustadt@csc.liv.ac.uk

Abstract. In this paper we give an overview of results for modal logic which can be shown using techniques and methods from first-order logic and resolution. Because of the breadth of the area and the many applications we focus on the use of first-order resolution methods for modal logics. In addition to traditional propositional modal logics we consider more expressive *PDL*-like dynamic modal logics which are closely related to description logics. Without going into too much detail, we survey different ways of translating modal logics into first-order logic, we explore different ways of using first-order resolution theorem provers, and we discuss a variety of results which have been obtained in the setting of first-order resolution.

1 Introduction

The main motivation for reducing problems in one logic (the source logic) to 'equivalent' problems in another logic (the target logic) is to exploit results of the target logic to draw some conclusions about the initial problems and use existing methods and tools of the target logic for the purpose of solving problems in the source logic. Reduction of modal logic problems to first-order logic is the pertinent case considered in this paper. There are good reasons for following this approach. First, a plethora of results on first-order logic and subclasses of it are available, including (un)decidability results, complexity results, correctness results for a wide range of calculi for first-order logic, and results on practical aspects and optimisation of the implementation of these calculi. Second, over the years a number of first-order logic theorem provers have been developed,

[⋆] For us there was no doubt that the topic of this chapter should be about automated reasoning and specifically resolution. We could have made this an overview paper of automated reasoning and resolution for solvable first-order classes, for description logics or for modal logics. We decided on an overview of automated reasoning and resolution for modal logics because much of the research carried out in Harald Ganzinger's research group (to which we had the privilege of belonging from 1991 to 1997/8) has been in this area. In addition, because of the close relationship between modal logics and description logics all of the results and observations about modal logics immediately carry over to description logics. Manuscript of (2006), to appear in Podelski, A., Voronkov, A. and Wilhelm, R. (eds), Volume in memoriam of Harald Ganzinger. *Lecture Notes in Artificial Intelligence*, Springer.

and the current generation of provers have reached a high level of sophistication and have been extensively tuned for efficiency. These two points together make first-order logic a natural choice as a target logic, in particular, for source logics reducible to first-order logic. A wide range of non-classical logics, either in the guise of modal logics or description logics, fall into this category. For these logics, the advantage of using a translation approach is that no major implementation effort is necessary, most often all that is required is the implementation of translation routines. This contrasts with special-purpose approaches for which either an implementation needs to be developed from scratch or requires an existing implementation to be adapted and extended for each new logic. In the case of modal logic a number of translation approaches are already implemented in MSPASS [54, 83] so that in this case no implementation overhead is needed, because MSPASS can be used directly as a modal logic theorem prover.

There are however not only practical advantages to using translation-based approaches. There are also significant theoretical advantages. By using translations it is often possible to transfer results and insights about one logic to another logic, similarly it is possible to transfer knowledge about certain classes of problems to other classes of problems. For example, in logic and computer science decidability and undecidability results are often shown by methods of translation, then usually referred as methods of interpretation or reduction methods. Complexity theory is another example where translations play a key role. In this case reductions are used for obtaining worst-case complexity results. This allows for the classification of decidable problems into broad complexity classes. Mathematicians and logicians usually just use interpretation methods and reductions as tools for proving, say, decidability, undecidability or complexity results, but are not normally interested in developing algorithms or proof procedures which could exploit these methods. Often the reductions are indeed not practically feasible, but when they are constructive then there is a possibility that these can be turned into efficient, implemented algorithms. There are few examples where such possibilities have been seriously explored. It is therefore perhaps not surprising that some authors are put off by the idea of using a translation approach for automated deduction. Such attitudes are however completely unjustified. From the perspective of computer science, adopting the translation approach is like using compilers to convert programs into machine code instructions for execution (or like converting text files into dvi files). It is generally acceptable to use compilers and interpreters in programming and we see no reason why the use of translation approaches for reasoning should not be equally acceptable. If we think about it then all inference steps of any calculus are just transformation steps anyway.

In this paper it is our intention to give an overview of some of the many uses of translation approaches and techniques at the heart of automated reasoning. Over the years a powerful and versatile framework of first-order resolution has evolved, which is increasingly applied to obtain strong results in numerous fields, both inside as well as outside, of automated reasoning. Recent results suggest that the modern framework of (first-order) resolution provides perhaps the most successful, current approach for developing practical decision procedures, certainly within the scope of first-order logic. Very many modal logics and expressive description logics are decidable by resolution. Moreover, the same refinements of resolution that decide these logics can be seen to decide very expressive fragments of first-order logic (guarded fragments, the two-variable fragment, fluted logic, Maslov's class K, the Bernays-Schönfinkel class). It can be shown that there is a one-to-one correspondence between most tableau methods for modal logics, including description logics, and a certain kind of first-order resolution method. This simulation by resolution has a number of fundamental consequences. It provides valuable insight into the similarities and differences of resolution and tableau methods. Since for the present day modal logic and description logic systems simulations in the other direction are currently absent this also shows that resolution-based methods are very powerful, and in many cases provably more powerful. Simulation results show a close connection between different styles of proof methods and can also be used as a basis for systematically developing specialised reasoning methods. In this way it has been possible to develop completely new tableau-like calculi for expressive modal and description logics within the resolution framework. Many results which are traditionally proved using model-theoretical constructions can in fact be proved within the resolution framework (soundness and completeness of special-purpose proof systems, finite model property, correspondence properties, interpolation, etc). First-order logic theorem proving tools are therefore not only useful for solving reasoning problems in modal logics, but can be exploited for realising and testing theoretical results about modal logics. In contrast to special-purpose approaches no major implementation effort is necessary; often all that is required is the implementation of translation routines. Decision procedures and simulations of other styles of deduction can then be obtained by simply selecting a correct set of parameters for the first-order prover. All this makes the combination of the translation approach and resolution very attractive for solving all kinds of problems requiring reasoning within and about modal logics (but not just modal logics).

The purpose of this overview paper is to show the application of first-order resolution methods for a large class of modal logics. In addition to traditional propositional modal logics we also consider more expressive *PDL*-like dynamic modal logics which are closely related to description logics. We review a variety of the different uses of first-order resolution for these logics. In particular, we focus on the development of practical inference methods, for studying specific issues such as decidability, the finite model property, the automatic generation of models, automated correspondence theory, and the relationships to other deduction approaches such as tableaux.

The paper is structured as follows. In Section 2 we recall standard definitions of resolution-based first-order theorem proving. Section 3 defines the essential concepts of the modal logics considered in this paper, namely syntax, the standard relational semantics and the corresponding relational translation method. These are defined for two kinds of modal logics: traditional style modal logics (Section 3.1) and dynamic modal logics (Section 3.2). Section 4 describes numerous alternative translations methods which are classified as syntactic approaches (Section 4.1), semantic approaches (in particular, approaches based on functional translations, Section 4.2), and approaches combining syntactic and semantic elements (in particular, the axiomatic translation approaches, Section 4.3). Section 5 reviews solvable first-order fragments and solvable clausal classes to which modal logics can be mapped using these different translation approaches. In particular, fragments and classes relevant to the relational translation (Section 5.2), the semi-functional translation (Section 5.3), the optimised functional translation (Section 5.4), and the axiomatic translation methods (Section 5.5) are discussed in turn. The topic of Section 6 is decision procedures. Various kinds of resolution decision procedures ranging from decision procedures not relying on any kind of refinements (Section 6.1), decision procedures based on ordering refinements (Section 6.2) and selection-based refinements (Section 6.3) are considered. Section 7 focusses on the connection between resolution and tableaux, in particular, how this connection has been exploited to simulate, develop and study modal tableau proof approaches. Resolution-based methods are also suitable for automatically generating Herbrand models. The application of such methods to modal logics is discussed in Section 8. Section 9 discusses resolution-based algorithms capable of transforming modal logic axioms into their corresponding first-order frame properties. Finally, in the Conclusion we give pointers to survey papers on related topics and summarise the direct contributions of Harald Ganzinger to modal logic theorem proving.

2 The modern resolution framework

The resolution calculus operates on sets of clauses. Clauses are quantifier free disjunctions of literals which may contain function symbols. The variables are implicitly assumed to be universally quantified. Any first-order formula can be transformed into an satisfiability equivalent set of clauses.

Theorem 1. *There is a polynomial or linear reduction* Cls *of any first-order formula to clause logic such that φ valid in first-order logic iff* $\mathrm{Cls}(\neg\varphi)$ *is unsatisfiable.*

The clausal form is obtained by transformation to conjunctive normal form, Skolemisation and crucially involves structural transformation which introduces new predicate symbols and definitions. Since resolution is a refutation calculus, instead of proving theoremhood, resolution attempts to refute the negation of a given formula.

The basic (unrefined) resolution calculus is very simple and consists of two inference rules, the resolution rule and the factoring rule, and no axioms. For propositional logic the resolution rule is just the operation that infers a clause $C \vee D$ from two clauses $C \vee A$ and $D \vee \neg A$. The factoring rule is a form of simplification which eliminates multiple copies of the same literal from one clause, that is, it infers $C \vee A$ from $C \vee A \vee A$. These two rules suffice to give us a sound and complete calculus for propositional logic and sets of ground clauses. We obtain a sound and complete inference system for full first-order logic and clause logic, if we augment the rules with unification. This calculus, the *basic resolution calculus*, due to Robinson [79] is sound and complete for full first-order logic and clause logic. It is however hopelessly inefficient. This was already noticed in the very early stages of the development of first-order resolution methods (by Robinson and others). The first papers on refinements of resolution appeared in the same year that Robinson published his famous paper which introduced resolution. Since the mid-sixties the advances have been impressive. The current generation of theorem provers, which include SPASS [93], E [89] and VAMPIRE [77] (in order of creation), are based on the modern framework of resolution and superposition to whose development Harald Ganzinger has made important contributions. In the following, when we refer to *resolution* we mean the resolution (and superposition) framework described in a series of papers by Bachmair and Ganzinger [5–7], see also, e.g. [62].

The main ingredients of this framework are refinements of the inference rules which restrict their applicability and a general notion of redundancy. Refinements of inference rules are defined in terms of two parameters: an ordering $\succ$ and a selection function S. The idea is that inferences do not need to be performed (but can) unless they are on literals maximal under the given ordering or on (negative) literals selected by the selection function S. The selection function can override the ordering. That is, if a literal is selected then it is the preferred candidate for an inference step even though there may be 'larger' literals in the clause. The ordering and selection function are used to limit the number of possible inferences. It is clear that, in general, if we can reduce the number of possible inferences without losing completeness then a proof can be found more quickly as the search space for the proof is reduced. There is a general completeness proof due to Bachmair and Ganzinger which requires only weak conditions for the admissibility of orderings and selection functions.

Simplification and deletion rules are important regardless of the style of deduction one uses. In the resolution framework these are based on a general notion of redundancy, which is based on considerations of the model construction which is at the centre of the completeness proof of the framework. Standard simplification rules like elimination of duplicate literals within a clause, tautology deletion, subsumption deletion (forward and backward subsumption deletion), condensing, etc, are instances of this notion [7].

Let $\mathsf{R}^{\mathrm{red}}_{\mathrm{sp}}$ be the resolution calculus defined by the rules of Figure 1. (The meaning of 'red' in the notation is 'with redundancy' and the meaning of 'sp' is 'with splitting'.) In our presentation we distinguish four kinds of rules. The Deduce rules are the ordered resolution and positive factoring rules. The ordering $\succ$ is a parameter which can be any admissible ordering and S is any selection function of negative literals. The Delete and Simplify rules are deletion and replacement rules compatible with the general notion of redundancy of [7, 8]. Essentially, a ground clause is redundant with respect to a set N and the ordering $\succ$ if it follows from smaller instances of clauses in N, and a non-ground clause is redundant in N if all its ground instances are redundant in N.

Deduce: $$\frac{N}{N \cup \{C\}}$$ if C is a factor or resolvent of premises in N.

Delete: $$\frac{N \uplus \{C\}}{N}$$ if C is redundant with respect to N.

Simplify: $$\frac{N}{(N \backslash M) \cup M'}$$ if $(N \backslash M) \cup M'$ is satisfiable when N is satisfiable and every clause in M is redundant with respect to $(N \backslash M) \cup M'$.

Split: $$\frac{N \uplus \{C \vee D\}}{N \cup \{C\} \mid N \cup \{D\}}$$ if C and D are variable-disjoint.

Resolvents and factors are computed with:

Ordered resolution: $$\frac{C \vee A \quad \neg B \vee D}{(C \vee D)\sigma}$$

provided (i) σ is the most general unifier of A and B, (ii) no literal is selected in C, and $A\sigma$ is strictly $\succ$-maximal with respect to $C\sigma$, and (iii) $\neg B$ is either selected, or $\neg B\sigma$ is maximal with respect to $D\sigma$ and no literal is selected in D.

Ordered factoring: $$\frac{C \vee A \vee B}{(C \vee A)\sigma}$$

provided (i) σ is the most general unifier of A and B, and (ii) no literal is selected in C and $A\sigma$ is $\succ$-maximal with respect to $C\sigma$.

Fig. 1. The calculus $\mathcal{R}^{\text{red}}_{\text{sp}}$

Tautology deletion is a familiar instance of the Delete rule. Other examples are the forward and backward subsumption deletion rules. A clause D *subsumes* a clause C iff there exists a substitution σ such that $D\sigma \subseteq C$ (strictly speaking, in the framework $D\sigma \subset C$ has to hold). Condensing is an instance of the Simplify rule. The *condensation* cond(C) of a clause C is a minimal multiple (positive or negative) factor of C which subsumes C (minimality is with respect to the number of literals in the clause). A clause C is *condensed* if there is no proper subclause of C which is a factor of C. Testing for redundancy in its general form is an expensive operation; in first-order logic general redundancy elimination is undecidable. For this reason one does not find theorem provers that implement redundancy elimination in the full generality. Only effectively computable instances of the Delete and Simplify rules are normally implemented in theorem provers. The Split rule is a rule familiar from DPLL algorithms and tableau calculi. Instead of refuting $N \cup \{C \vee D\}$ one refutes both $N \cup \{C\}$ and $N \cup \{D\}$ [15] (alternatively, it is possible to use the complement splitting rule, which means that instead of refuting $N \cup \{C \vee D\}$ one refutes both $N \cup \{C\}$ and $N \cup \{\neg C, D\}$). The splitting rule is don't know non-deterministic and usually requires backtracking. However, in the resolution framework splitting can be simulated by transformation steps which introduce new propositional symbols. If $C \vee D$ is a clause that can be split into two split components C and D, then

Ordered hyperresolution: $$\frac{C_1 \vee A_1 \quad \ldots \quad C_n \vee A_n \quad \neg B_1 \vee \ldots \vee \neg B_n \vee D}{(C_1 \vee \ldots \vee C_n \vee D)\sigma}$$

provided (i) σ is the most general unifier such that $A_i\sigma = B_i\sigma$ for every i, $1 \leq i \leq n$, (ii) $A_i\sigma$ is strictly $\succ$-maximal with respect to $C_i\sigma$, and the C_i are positive clauses, for every i, $1 \leq i \leq n$, and (iii) for every i, $1 \leq i \leq n$, $\neg B_i$ is selected and D is a positive clause.

Ordered factoring: $$\frac{C \vee A \vee B}{(C \vee A)\sigma}$$

provided (i) σ is the most general unifier of A and B, and (ii) C is positive and $A\sigma$ is $\succ$-maximal with respect to $C\sigma$.

Fig. 2. The Deduce rules of ordered hyperresolution

it is possible to replace $C \vee D$ by two clauses $C \vee q$, and $\neg q \vee D$. The new symbol q is made minimal in the ordering $\succ$, and $\neg q$ is selected [19, 78]. In most cases this is easier to implement than the full splitting rule, but has also other advantages.

The calculus without the splitting rule is denoted by $\mathsf{R}^{\mathrm{red}}$ and R is the calculus with just Deduce rules.

Theorem 2 ([7, 8]). *(i) $\mathsf{R}^{\mathrm{red}}_{\mathrm{sp}}$ is a sound and complete refutation system for clause sets. (ii) $\mathsf{R}^{\mathrm{red}}$ (no splitting) is a sound and complete refutation system for clause sets.*

Corollary 1. *R is a sound and complete refutation system for clause sets.*

The *(ordered) hyperresolution calculus* is based on maximal selection of negative literals. This means the selection function selects exactly the set of all negative literals in any non-positive clause. Let $\mathsf{OH}^{\mathrm{red}}_{\mathrm{sp}}$ be the calculus based on maximal selection and an ordering $\succ$, where the Deduce rules are given by the rules in Figure 2. This means the rules are the above hyperresolution rule, positive factoring, redundancy elimination and splitting. Similar as above, $\mathsf{OH}^{\mathrm{red}}$, respectively OH, denotes the calculus $\mathsf{OH}^{\mathrm{red}}_{\mathrm{sp}}$ but without the splitting rule, respectively the calculus just consisting of Deduce rules. For completeness an ordering refinement is optional. We use the notation $\mathsf{H}^{\mathrm{red}}_{\mathrm{sp}}$, $\mathsf{H}^{\mathrm{red}}$, and H for the unordered versions.

Corollary 2. *$\mathsf{OH}^{\mathrm{red}}_{\mathrm{sp}}$, $\mathsf{OH}^{\mathrm{red}}$, OH, $\mathsf{H}^{\mathrm{red}}_{\mathrm{sp}}$, $\mathsf{H}^{\mathrm{red}}$ and H are sound and complete refutation systems for clause sets.*

3 Modal logic and the classical relational translation

We are interested in a broad class of modal logics. These include the *traditional modal logics* which have been extensively studied since the 1960ies and even well

	Axiom $\mathcal{A}$	Correspondence property	
T	$\Box_i p \rightarrow p$	reflexivity	$\forall x\, R_i(x,x)$
B	$\Diamond_i \Box_i p \rightarrow p$	symmetry	$\forall xy\, (R_i(x,y) \rightarrow R_i(y,x))$
D	$\Box_i p \rightarrow \Diamond_i p$	seriality	$\forall x \exists y\, R_i(x,y)$
4	$\Box_i p \rightarrow \Box_i \Box_i p$	transitivity	$\forall xyz\, (R_i(x,y) \wedge R_i(y,z) \rightarrow R_i(x,z))$
alt_1	$\Diamond_i p \rightarrow \Box_i p$	functionality	$\forall xyz\, (R_i(x,y) \wedge R_i(x,z) \rightarrow y \approx z)$
5	$\Diamond_i \Box_i p \rightarrow \Box_i p$	Euclideanness	$\forall xyz\, (R_i(x,y) \wedge R_i(x,z) \rightarrow R_i(y,z))$

Fig. 3. Axioms and relational background theories

before then. There are also PDL-like modal logics, which we refer to as *dynamic modal logics*, because they have a dynamic component (i.e. relational operators) which traditional modal logics do not have. Dynamic modal logics are of particular interest because of the close relationship to description logics. Description logics are popular in the area of knowledge representation and have become a topic of much interest in connection with the semantic web and ontologies.

3.1 Traditional modal logics

The language of modal logic is an extension of the language of propositional logic with a family of unary modal operators $\Box_i$ (the box operators). More precisely, given a countably finite set of propositional variables $p, p_1, p_2, \ldots$, modal formulae are defined inductively as follows. Every propositional variable is a modal formula. If ϕ and ψ are modal formulae, then so are $\neg\phi$, $\phi \wedge \psi$, and $\Box_i \phi$. Other familiar connectives such as $\bot$, $\top$, $\vee$, $\rightarrow$ and $\leftrightarrow$ can then be defined as expected. As usual we let $\Diamond_i \phi =^{\mathrm{def}} \neg\Box_i\neg\phi$. A *substitution* σ is a mapping from propositional variables to modal formulae which can be lifted to modal formulae in the standard way. The application of σ to a modal formula ϕ is denoted by $\phi\sigma$ and $\phi\sigma$ is called an *instance* of ϕ. A *modal axiom (schema)* is a modal formula ϕ representing the set of all instances of ϕ.

A *(normal) modal logic* is defined by a set of modal formulae which includes all propositional tautologies and the axiom $\Box_i(p \rightarrow q) \rightarrow (\Box_i p \rightarrow \Box_i q)$, called the axiom K, for each $\Box_i$ modality in the language. In addition this set is closed under the rule of uniform substitution (if $\vdash p$ and σ is a substitution, then $\vdash p\sigma$), modus ponens (if $\vdash p$ and $\vdash p \rightarrow q$ then $\vdash q$) and the rule of necessitation (if $\vdash p$ then $\vdash \Box_i p$). $K_{(m)}$ is the weakest multi-modal logic, that is, the logic given by the smallest set of modal formulae constituting a normal modal logic. Other modal logics commonly considered in the literature and used in applications are extensions of the logic $K_{(m)}$ with additional axioms such as those listed in Figure 3. In this paper we refer to these logics as *traditional modal logics* to distinguish them from the dynamic modal logics considered in the next section. Let Δ denote a finite set of formulae. By $K_{(m)}\Delta$ we denote the smallest modal logic extending $K_{(m)}$ which includes all substitution instances of formulae in Δ (the formulae in Δ are the extra axioms) and is closed under the inference rules. A modal formula φ is a *theorem* of $K_{(m)}$ or one of its extensions iff φ can be derived by using the axioms and the rules of the logic.

The standard semantics of propositional modal logics, known as the *Kripke semantics* or *possible world semantics*, is given in terms of relational structures called *frames*. A frame of a modal logic is a pair $\mathcal{F} = (W, \{R_i\}_i)$ of a non-empty set of worlds W and a family of binary accessibility relations R_i over W. The defining class of frames of a modal logic determines, and is determined by, a corresponding class of models. A (*relational*) *model* is a pair $\mathcal{M} = (\mathcal{F}, v)$ of a frame $\mathcal{F}$ and a *valuation* function v. v assigns subsets of W to atomic propositional variables. The model $\mathcal{M}$ is therefore said to be *based on the frame $\mathcal{F}$*. *Truth* in any model $\mathcal{M} = (W, \{R_i\}_i, v)$ and any world $x \in W$ is defined inductively by:

$$\mathcal{M}, x \models p \text{ iff } x \in v(p) \qquad\qquad \mathcal{M}, x \models \neg\phi \text{ iff } \mathcal{M}, x \not\models \phi$$
$$\mathcal{M}, x \models \phi \wedge \psi \text{ iff both } \mathcal{M}, x \models \phi \text{ and } \mathcal{M}, x \models \psi$$
$$\mathcal{M}, x \models \Box_i\phi \text{ iff } (x, y) \in R_i \text{ implies } \mathcal{M}, y \models \phi, \text{ for any } y \in W$$

If $\mathcal{M}, x \models \varphi$ holds then we say φ is true at x in $\mathcal{M}$ and that $\mathcal{M}$ *satisfies* φ. A modal formula φ is *satisfiable* iff there exists a model $\mathcal{M}$ and a world x in $\mathcal{M}$ such that $\mathcal{M}, x \models \varphi$. A modal formula is *valid in a frame* iff it is valid in all models based on the frame. The basic multi-modal logic $K_{(m)}$ is completely determined by the class of all frames.

Normal modal logics can be studied systematically by considering the classes of frames they define. In general, these are subclasses of the class of all frames which define the basic modal logic $K_{(m)}$. A modal logic $K_{(m)}\Delta$ is said to be *sound* (respectively *complete*) *with respect to a class of frames* iff for any modal formula φ, any frame in the class validates φ if (respectively iff) φ is a theorem in $K_{(m)}\Delta$. A modal logic is said to be *complete* iff it is complete with respect to some class $\mathfrak{F}$ of frames.[3] Given a modal logic $K_{(m)}\Delta$ and a class $\mathfrak{F}$ of frames for which it is sound and complete, $\mathfrak{F}$ can be characterised by a second-order formula Ψ, that is, there is a second-order formula Ψ such that a frame $\mathcal{F}$ is in $\mathfrak{F}$ iff Ψ is true for $\mathcal{F}$. In some cases, but not always, the second-order formula Ψ is equivalent to a first-order formula ψ and ψ is then called a *first-order correspondence property* for $K_{(m)}\Delta$. A class of frames comprising of all frames satisfying a set of first-order conditions is an *elementary* class. Figure 3 lists the first-order correspondence properties satisfied by classes of frames for extensions $K_{(m)}\Delta$ for a selection of common axioms.

When given a modal formula φ, the problem of determining whether there exists a model $\mathcal{M}$ and a world x in $\mathcal{M}$ such that φ is true at x in $\mathcal{M}$ is also called the *local satisfiability problem*. By contrast, the *global satisfiability problem* is the problem of determining whether there exists a model $\mathcal{M}$ such that for every world x in $\mathcal{M}$, the given φ is true at x in $\mathcal{M}$.

We are also interested in inference problems relative to finite sets of assumptions or *background theories*. We use the notation Γ for background theories. The modal formulae in Γ are referred to as *non-logical axioms*, because by definition they are formulae true in every world of a model but are not logical axioms which

[3] Note in modal logic the notion of completeness is used differently than in other logical disciplines.

are required to be true in every world in every model of a class. A model $\mathcal{M}$ *(globally) satisfies* a background theory $\varGamma$ if, for each world x in $\mathcal{M}$ and each $\rho \in \varGamma$, it is the case that $\mathcal{M}, x \models \rho$. We can then consider the local satisfiability of a modal formula φ with respect to a background theory $\varGamma$, i.e., whether there exists a model $\mathcal{M}$ and a world x in this model so that φ is true in x and $\varGamma$ is globally true in $\mathcal{M}$. Global satisfiability with respect to a background theory is defined as expected.

The standard interpretation of (first-order definable) modal logics inside first-order logic is given by the *classical (relational) translation* mapping, here denoted by π_r. It maps modal formulae to first-order formulae by uniquely associating each propositional variable p with unary predicate symbol Q_p and simply mimicking the semantic definitions of the operators.

$$\begin{array}{ll} \pi_r(p,x) = Q_p(x) & \pi_r(\phi \wedge \psi, x) = \pi_r(\phi, x) \wedge \pi_r(\psi, x) \\ \pi_r(\neg\phi, x) = \neg\pi_r(\phi, x) & \pi_r(\Box_i \phi, x) = \forall y\, (R_i(x,y) \rightarrow \pi_r(\phi, y)) \end{array}$$

Let $\mathrm{Corr}(\varDelta)$ denote the set of first-order frame properties of the axioms in $\varDelta$.

Theorem 3. *Let L be a first-order definable propositional multi-modal logic $K_{(m)}\varDelta$ which is complete with respect to the class of frames satisfying* $\mathrm{Corr}(\varDelta)$*. Then, for any formula φ and any finite set $\varGamma$ of non-logical axioms, (i) π_r can be computed in linear time, (ii) φ is locally satisfiable in L iff* $\mathrm{Corr}(\varDelta) \wedge \bigwedge_{\rho\in\varGamma} \forall x\, \pi_r(\rho, x) \wedge \exists x\, \pi_r(\varphi, x)$ *is satisfiable in first-order logic, and (iii) φ is globally satisfiable in L iff* $\mathrm{Corr}(\varDelta) \wedge \bigwedge_{\rho\in\varGamma} \forall x\, \pi_r(\rho, x) \wedge \forall x\, \pi_r(\varphi, x)$ *is satisfiable in first-order logic.*

3.2 Dynamic modal logics

Dynamic modal logics provide another kind of extension of the basic modal logic $K_{(m)}$. They are *PDL*-like modal logics in which the modal operators are parameterised by relational formulae. These can be used to formalise dynamic notions such as actions or programs and are useful in linguistic or AI applications. Logics which belong to this family of logics, or are closely related, are propositional dynamic logic *PDL* itself, but also Boolean modal logic *BML* [34], tense logic, information logics [23], dynamic modal logic [21], logics expressing inaccessibility and sufficiency [35, 47, 48] as well as a large class of description logics.

The strongest logic in the class of dynamic modal logics we consider in this paper is called *Peirce logic*. Peirce logic subsumes all the logics mentioned in the previous paragraph except for *PDL* (but can be easily extended to also subsume *PDL*). Peirce logic is a logical formalisation of representable Peirce algebras [60, 88] and is closely related to de Rijke's dynamic modal logic [21].

Formally, Peirce logic is the modal logic defined over relations which form a relation algebra. The language of Peirce logic consists of two syntactic types: dynamic modal formulae and relational formulae. The logical connectives are (i) the connectives of the modal logic $K_{(m)}$, with the difference that the modal operators are indexed with relational formulae, instead of just numbers, (ii) the standard connectives of relational logics, namely: ; (composition), $\smile$ (converse), *id* (identity), and (iii) a left cylindrification operator ^{c}. (Instead of the left cylindrification operator one could have chosen the test operator of *PDL*, domain restriction, range restriction, or cross product, cf. [16]; the symbol *id* is a logical constant which is interpreted as the identity relation.) Given countably many propositional variables denoted by p_j as before, and countably many relational variables, denoted by r_i, *dynamic modal formulae* and *relational formulae* are defined inductively as follows. Every propositional variable is a dynamic modal formula and every relational variable is a relational formula. If ϕ, ψ are dynamic modal formulae and α, β are relational formulae, then $\neg\phi$, $\phi \wedge \psi$, $[\alpha]\phi$ are dynamic modal formulae and *id*, $\neg\alpha$, $\alpha \wedge \beta$, $\alpha\,;\beta$, $\alpha^{\smile}$, ϕ^c are relational formulae.

We define the set of *formulae* of Peirce logic to be the set of dynamic modal formulae. The definition in [88] allows also relational formulae as first-class citizens, but these can be expressed in terms of dynamic modal formulae (see below). This variation in the definition is thus inconsequential.

The semantics of Peirce logic is defined in terms of frames, where a frame is a tuple (W, R) of a non-empty set W (of worlds) and a mapping R from relational formulae to binary relations over W satisfying:

$$\begin{array}{lll} R_{\neg\alpha} = W^2 \backslash R_\alpha & R_{\alpha\wedge\beta} = R_\alpha \cap R_\beta & R_{\alpha\,;\,\beta} = R_\alpha\,; R_\beta \\ R_{\alpha^\smile} = R_\alpha^\smile & R_{id} = Id_W & R_{\phi^c} = \{(x,y) \in W^2 \mid x \in v(\phi)\}. \end{array}$$

Here and in the rest of the paper we prefer to use the notation R_α instead of $R(\alpha)$. Id_W denotes the identity relation on the set W, while $R^\smile$ denotes the converse (or inverse) of a relation R. A model is now given by a triple $\mathcal{M} = (W, R, v)$, where (W, R) is a frame and v is a mapping from propositional variables to subsets of W satisfying the same conditions as before with the following change.

$$\mathcal{M}, x \models [\alpha]\phi \text{ iff } (x,y) \in R_\alpha \text{ implies } \mathcal{M}, y \models \phi, \text{ for any } y \in W$$

The definitions of the notions of satisfiability and validity remain the same. The notions of local satisfiability, global satisfiability and satisfiability with respect to a set of non-logical axioms are also defined as for traditional modal logics.

There is a natural hierarchy of dynamic modal logics in which $K_{(m)}$ forms the weakest logic and Peirce logic forms the strongest logic. It is not difficult to see that $K_{(m)}$ is just the reduct of Peirce logic in which relational formulae are limited to m relational variables r_j. *Dynamic modal logics* are defined over the language of $K_{(m)}$ but include in addition to the operators of $K_{(m)}$ and relational variables also finitely many relational operators $\star_1, \ldots, \star_k$. We assume that these relational operators are definable in terms of the operators of Peirce logic. A logic $K_{(m)}(\star_1, \ldots, \star_k)$ is then defined to be the multi-modal logic defined over relations closed under the set-theoretic operations corresponding to the relational operators $\star_1, \ldots, \star_k$. The class of dynamic modal logics forms therefore a lattice in which $K_{(m)}$ is the weakest logic and every other logic is obtained by enhancing the language with one or more relational operators $\star_i$. Clearly some of the logics in this lattice are expressively equivalent. Peirce logic

Property of R_r	Relational formula	Property of R_r	Relational formula
reflexivity	$id \rightarrow r$	transitivity	$r\,;r \rightarrow r$
symmetry	$r \rightarrow r^{\smile}$	functionality	$r^{\smile}\,;r \rightarrow id$
seriality	$\top \rightarrow r\,;\top$ or $\top \rightarrow \langle r\rangle\top$	Euclideanness	$r\,;r^{\smile} \rightarrow r$

Fig. 4. Relational properties expressed as relational formulae. (The relational operators are assumed to have higher priority than →.)

is equivalent to the top element in the lattice. Tense logic coincides with the logic $K_{(m)}(^{\smile})$, *BML* coincides with $K_{(m)}(\neg,\wedge)$, and the description logic $\mathcal{ALB}$ [53] coincides with $K_{(m)}(\neg,\wedge,^{\smile},\upharpoonleft)$. The operator $\upharpoonleft$ is the domain restriction operator, i.e. $\alpha\upharpoonleft\phi =^{\mathrm{def}} \alpha \wedge \phi^c$. Range restriction may be defined by $\alpha \downharpoonleft \phi =^{\mathrm{def}} (\alpha^{\smile}\upharpoonleft\phi)^{\smile}$.

Relational formulae as 'independent' formulae are implicit in all dynamic modal logics with relational negation and relational conjunction or disjunction. For example, implication between relational formulae can be defined by $(\alpha \rightarrow \beta) =^{\mathrm{def}} \top \rightarrow [\alpha \wedge \neg\beta]\bot$. Then it is also possible to specify properties of the underlying accessibility relations. Figure 4 gives some properties of the accessibility relation associated with r which can be specified by relational formulae in Peirce logic. If r is a relational variable then the universal modality can be defined by either $[\top]\phi =^{\mathrm{def}} [r \vee \neg r]\phi$ or $[\top]\phi =^{\mathrm{def}} [\neg(r \wedge \neg r)]\phi$. The test operator of *PDL* can be defined by $\phi? =^{\mathrm{def}} id \wedge \phi^c$. We also define the operator $id(\cdot)$ by $id(\alpha) =^{\mathrm{def}} \alpha \wedge id$.

The standard (relational) translation of Peirce logic into first-order logic is specified by the following.

For dynamic modal formulae:

$$\pi_r(p,x) = Q_p(x) \qquad \pi_r(\phi \wedge \psi, x) = \pi_r(\phi,x) \wedge \pi_r(\psi,x)$$

$$\pi_r(\neg\phi,x) = \neg\pi_r(\phi,x) \qquad \pi_r([\alpha]\phi,x) = \forall y\,(\tau(\alpha,x,y) \rightarrow \pi_r(\phi,y))$$

For relational formulae:

$$\tau(r,x,y) = Q_r(x,y) \qquad \tau(\alpha \wedge \beta,x,y) = \tau(\alpha,x,y) \wedge \tau(\beta,x,y)$$

$$\tau(\neg\alpha,x,y) = \neg\tau(\alpha,x,y) \qquad \tau(\alpha\,;\beta,x,y) = \exists z\,(\tau(\alpha,x,z) \wedge \tau(\beta,z,y))$$

$$\tau(\alpha^{\smile},x,y) = \tau(\alpha,y,x) \qquad \tau(\phi^c,x,y) = \pi_r(\phi,x)$$

$$\tau(id,x,y) = x \approx y$$

Theorem 4. *Let L be a logic in the lattice of dynamic modal logics and let Σ be a (possibly empty) set of first-order relational frame properties. Then, for any formula φ, (i) $\pi_r(\varphi,x)$ can be computed in linear time, and (ii) φ is satisfiable in L with respect to Σ iff $\Sigma \wedge \exists x\,\pi_r(\varphi,x)$ is first-order satisfiable.*

4 Different translation methods

The standard translation of propositional modal logics and dynamic modal logics presented in the previous section is just one among many other reductions of these logics to first-order logic. A variety of non-standard translation mappings have been developed and studied. Experience shows that the performance of a first-order logic prover depends heavily on the translation chosen as well as subsequent transformations. Choosing a suitable translation mapping for a particular modal logic is therefore an important decision to make.

Broadly there are three types of translation methods: (i) syntactic translation methods, (ii) semantic translation methods, and (iii) a mixture of both.

4.1 Syntactic translation

If a sound and complete axiomatisation exists for a logic then it is normally straightforward to interpret the logic inside first-order logic. Take for example the Hilbert-style axiomatisations of the modal logic *S4*. This consists of an axiomatisation of propositional logic, the axioms K, T and *4*, and uniform substitution, modus ponens and necessitation. This can be straightforwardly embedded in first-order logic by encoding formulae as terms and using a special T predicate representing the truth (entailment) relation. Each axiom $\mathcal{A}$ is then encoded as an universally quantified formula of the form $\forall \overline{p}\, T(\pi_t(\mathcal{A}))$, for example the axiom *4* is encoded as the following term, $\pi_t(\mathit{4}) = i(b(p), b(b(p)))$, where i and b are designated function symbols representing implication and box. The rules are encoded as $\forall p_1 p_2\, ((T(p_1) \wedge T(i(p_1, p_2))) \rightarrow T(p_2))$ for modus ponens and $\forall p\, (T(p) \rightarrow T(b(p)))$ for necessitation. This kind of *syntactic translation* is quite old and is regularly used by Wos, McCune and others in their investigations into the existence of small axiom bases for algebras and other mathematical problems. Syntactic translations, or T*-encodings* as Ohlbach [69] calls them, are quite widely applicable. For instance, in contrast to semantics-based encodings they are readily applicable to a great many modal logics and algebras including second-order (modal) logics for which no first-order semantic characterisations are known.

4.2 Semantic translations

In modal logic *semantic translations* are more often used than syntactic translations. The main reason for this is that it is currently easier to control the theorem proving processes and reduce the search space of a theorem prover, and it is easier and more natural to prove interesting properties of the logics. In the previous section we gave the definitions of the classical translation of modal logics into first-order logic. Because the translation just follows the standard definition of the standard semantic definition in terms of of accessibility relations this translation is often referred to the *relational translation* approach.

Functional translations. The semantics of modal logics can also be specified as structures based on accessibility *functions* rather than accessibility relations. Corresponding translations to first-order logic, so-called *functional translations*, have been introduced and studied by a variety of authors, including [4, 27, 67, 82]. In this paper we want to highlight some results obtained in Saarbrücken and therefore confine ourselves to the semi-functional translation method due to Nonnengart [63] as well as the optimised functional translation method studied by Ohlbach and others [44, 72, 94].

Semi-functional translation. The semi-functional translation approach [63, 65] combines the advantages of the relational and functional translation approach and tries to avoid their disadvantages. For an elaboration of the considerations leading to the development of the semi-functional translation approach please refer to Nonnengart [64]. The translation is called *semi-functional* as it translates box modalities in the standard way, while diamond modalities are translated functionally. In the following we focus on the modal logic K and its extensions by a set of axiom schemas Δ, as in the case of just one modality, the semi-functional translation provides an elegant encoding of Δ which in turn allows us to obtain strong decidability results.

The semi-functional translation maps modal formulae to many-sorted first-order formulae. We distinguish between the sorts W and AF for worlds and accessibility functions. Unary predicate symbols have sort W, the binary predicate symbol R associated with the accessibility relation has sort $W \times W$, the constant symbol ϵ has sort W, and the binary (left-associative) function $[\cdot, \cdot]$ has sort $W \times AF \to W$. Then the semi-functional translation π_{sf} is defined as follows.

$$\begin{aligned} \pi_{sf}(p, s) &= Q_p(s) & \pi_{sf}(\phi \wedge \psi, s) &= \pi_{sf}(\phi, s) \wedge \pi_{sf}(\psi, s) \\ \pi_{sf}(\neg\psi, s) &= \neg\pi_{sf}(\psi, s) & \pi_{sf}(\Box\psi, s) &= \forall y(R(s, y) \to \pi_{sf}(\psi, y)) \\ & & \pi_{sf}(\Diamond\psi, s) &= \mathrm{def}(s) \wedge \exists\alpha\pi_{sf}(\psi, [s\alpha]) \end{aligned}$$

Note that $\forall y$ quantifies over a variable of sort W while $\exists\alpha$ quantifies over a variable of sort AF. The expression $[s\alpha]$ is of sort W. Since the semi-functional translation incorporates both the relational representation and the functional representation of the accessibility relation, it is necessary to relate the two representations by means of the following formula.

$$\mathrm{Sim}_{sf} = \forall x \forall\alpha(\mathrm{def}(x) \to R(x, [x\alpha])) \wedge \forall xy(R(x, y) \to \mathrm{def}(x))$$

The two conjuncts of Sim_{sf} are called the *simulator axioms* for R. The following theorem states that the translation preserves the satisfiability and unsatisfiability of modal formulae.

Theorem 5 (Nonnengart [64]). *Let $K\Delta$ be a complete modal logic with first-order definable relational frame properties* $\mathrm{Corr}(\Delta)$*. A modal formula φ in negation normal form is satisfiable iff $\pi_{sf}(\varphi, \epsilon) \wedge \mathrm{Sim}_{sf} \wedge \mathrm{Corr}(\Delta)$ is satisfiable.*

Note that for any modal formula φ, $\pi_{sf}(\varphi, \epsilon)$ only contains negative R-literals, that is, negative literals with predicate symbol R. Consequently, positive R-literals can only come from Sim_{sf} and from $\mathrm{Corr}(\Delta)$, the set of first-order frame properties associated with a set of axioms Δ. This allows us to compute the consequences $\mathcal{C}$ of Sim_{sf} and $\mathrm{Corr}(\Delta)$ without the need to take $\pi_{sf}(\varphi, \epsilon)$ into account.

K4	$\forall xy\forall\alpha((\mathrm{def}(x) \rightarrow R(x,[x\alpha])) \wedge ((\mathrm{def}(x) \wedge R(x,y)) \rightarrow R(x,[y\alpha])))$	*KD*	$\forall x\forall\alpha(\mathrm{def}(x) \wedge R(x,[x\alpha]))$
K45	$\forall xy\forall\alpha((\mathrm{def}(x) \rightarrow \mathrm{def}(y)) \wedge (\mathrm{def}(y) \rightarrow R(x,[y\alpha])))$	*KD45*	$\forall xy\forall\alpha(\mathrm{def}(x) \wedge R(x,[y\alpha]))$
KB	$\forall xy\forall\alpha((\mathrm{def}(x) \rightarrow \mathrm{def}(y)) \wedge (\mathrm{def}(x) \rightarrow R(x,[x\alpha])) \wedge (\mathrm{def}(x) \rightarrow R([x\alpha],x)))$	*KDB*	$\forall x\forall\alpha(\mathrm{def}(x) \wedge R(x,[x\alpha]) \wedge R([x\alpha],x))$
KD4	$\forall xy\forall\alpha(\mathrm{def}(x) \wedge R(x,[x\alpha]) \wedge (R(x,y) \rightarrow R(x,[y\alpha])))$	*KT*	$\forall x\forall\alpha(\mathrm{def}(x) \wedge R(x,x) \wedge R(x,[x\alpha]))$
S4	$\forall xy\forall\alpha(\mathrm{def}(x) \wedge R(x,x) \wedge (R(x,y) \rightarrow R(x,[y\alpha])))$	*S5*	$\forall xy(\mathrm{def}(x) \wedge R(x,y))$
K5	$\forall xy\forall\alpha\beta((\mathrm{def}(x) \rightarrow \mathrm{def}(y)) \wedge (\mathrm{def}(\epsilon) \rightarrow R(\epsilon,[\epsilon\alpha])) \wedge ((\mathrm{def}(x) \wedge \mathrm{def}(y)) \rightarrow R([x\alpha],[y\beta])))$	*KD5*	$\forall xy\forall\alpha\beta(\mathrm{def}(x) \wedge R(\epsilon,[\epsilon\alpha]) \wedge R([x\alpha],[y\beta]))$

Fig. 5. Logics and semi-functional frame properties

In fact, we are not interested in all consequences of Sim_{sf} and $\mathrm{Corr}(\Delta)$, but only in the subset $\mathcal{C}_R$ of positive R-literals of $\mathcal{C}$. Then, $\pi_{sf}(\varphi,\epsilon) \wedge \mathrm{Sim}_{sf} \wedge \mathrm{Corr}(\Delta)$ is satisfiable iff $\pi_{sf}(\varphi,\epsilon) \wedge \mathrm{Sim}_{sf} \wedge \mathcal{C}_R$ is satisfiable. This approach can be taken further. Instead of $\mathcal{C}_R$, which might be infinite, we may use any (finite) set of first-order formulae $\mathcal{C}'$ which has as logical consequences the same set $\mathcal{C}_R$ of positive R-literals, that is, $\pi_{sf}(\varphi,\epsilon) \wedge \mathrm{Sim}_{sf} \wedge \mathrm{Corr}(\Delta)$ is satisfiable iff $\pi_{sf}(\varphi,\epsilon) \wedge \mathrm{Sim}_{sf} \wedge \mathcal{C}'$ is satisfiable. The set $\mathcal{C}'$ can be seen as an alternative representation of $\mathcal{C}_R$. Figure 5 lists the formulae we obtain following this approach for a variety of well-known extensions of K by the axiom schemas *4*, *5*, B, D, and T. While the set of consequences of a set of first-order formulae can be computed automatically, for example, using resolution, there is as yet no way to automatically compute its alternative representation.

Theorem 6 (Nonnengart [64]). *Let Δ be a subset of $\{4,5,B,D,T\}$. Let $\mathrm{SF}(\Delta)$ be the first-order formula representing the frame properties of $K\Delta$ according to Figure 5. Then a modal formula φ in negation normal form is satisfiable iff $\pi_{sf}(\varphi,\epsilon) \wedge \mathrm{SF}(\Delta)$ is satisfiable.*

Optimised functional translation. The *(monadic) optimised functional translation* is as the name suggests an improvement of the functional translation. It maps the basic modal logic $K_{(m)}$ to many-sorted first-order logic, more precisely, to a monadic fragment of sorted first-order logic, called *basic path logic*, and maps extensions of $K_{(m)}$ with axioms to extensions of the basic path logic by equational theories [72, 82]. Basic path logic has a sort S_W for the set of worlds W and a sort S_i for each modality $\Box_i$ in the logic. For each i there is a binary, left-associative function $[\cdot,\cdot]_i$ of sort $S_W \times S_i \rightarrow W$. Also there are special unary predicates def_i of sort S_W representing subsets of W. Each propositional variable p is uniquely associated with a unary predicate symbols Q_p of sort S_W. Commonly, the (monadic) optimised functional translation π_{of} is defined as a two step process: (i) the application of the functional translation to a modal formula which translates it to basic path logic, followed by (ii) the application of a quantifier exchange operation which converts the first-order formula obtained from the functional translation into prenex normal form and moves all existential quantifiers outwards as far as possible (or inwards, depending on one's point of view). Since we focus here only on the satisfiability problem, we can give a simplified presentation of the optimised functional translation obtained in just one step.

$$\begin{array}{rlrl}
\pi_{of}(p,s) = Q_p(s) & & \pi_{of}(\varphi \wedge \psi, s) = \pi_{of}(\varphi, s) \wedge \pi_f(\psi, s) \\
\pi_{of}(\neg\varphi, s) = \neg\pi_{of}(\varphi, s) & & \pi_{of}(\Box_i\varphi, s) = \forall y{:}S_i(\mathrm{def}_i(s) \rightarrow \pi_{of}(\varphi, [s\,y{:}S_i]_i)) \\
 & & \pi_{of}(\Diamond_i\varphi, s) = \mathrm{def}_i(s) \wedge \pi_{of}(\varphi, [s\,y{:}S_i]_i)
\end{array}$$

The symbol s denotes a path and $y{:}S_i$ denotes a variable of sort S_i. The intuition of the term $[s\,y{:}S_i]_i$ is that it represents an i-successor world which is reached via the path s to its predecessor world followed by a y transition of type S_i. This means $[s\,y{:}S_i]_i$ represents both a world and the path via which it is reached from the initial world. The inclusion of a specification for diamond formulae in the above definition is intentional and so is the omission of the quantifiers. The optimised functional translation of a modal formula φ is given by $\pi_{of}(\varphi, x{:}S_W)$, where $x{:}S_W$ is an arbitrary variable of sort S_W, and $x{:}S_W$ as well as the $y{:}S_i$ from $\pi_{of}(\Diamond_i\varphi, s)$, are free variables which are implicitly existentially quantified. Similar as for the relational semantics, the functional semantics of extensions of $K_{(m)}$ can be characterised by frame properties, this time they are formulated over accessibility functions. Figure 6 gives a selection of modal axioms and equivalent first-order formulae of the functional frames. For some schemas (such as D, T and 4) these properties may be simplified by a form of globalisation; see [72, 82] for details.

Theorem 7 ([72]). *Let $K_{(m)}\Delta$ be a complete modal logic such that the functional frame properties corresponding to the axioms in Δ are a set Σ of first-order formulae. Then, a modal formula φ is satisfiable in $K_{(m)}\Delta$ iff the formula $\Sigma \wedge \exists x\overline{y}\,\pi_{of}(\varphi, x{:}S_W)$ is first-order satisfiable.*

An important advantage of the optimised translation is that a wider class of modal logics can be embedded into first-order logic than can be done with the relational translation method, cf. [72, 81].

If we are only interested in establishing the satisfiability of formulae in the basic modal logic $K_{(m)}$ or extensions of $K_{(m)}$ by the axiom D for some or all modalities, then the use of sorted first-order logic and binary function symbols can be avoided by using k-ary predicates where the sort information is coded into the predicate names [50], see also [45]. The k-ary predicate symbols are $Q_{p,\sigma}$ and $\mathrm{def}_{i,\sigma}$ where p denotes a propositional symbol, and σ is a sequence of length k

D	$\Box_i p \rightarrow \Diamond_i p$
	$\forall x{:}S_W \ \mathrm{def}_i(x{:}S_W)$
T	$\Box_i p \rightarrow p$
	$\forall x{:}S_W \ \exists y{:}S_i \ (\mathrm{def}_i(x{:}S_W) \wedge x{:}S_W \approx [x{:}S_W \, y{:}S_i])$
B	$p \rightarrow \Box_i \Diamond_i p$
	$\forall x{:}S_W \ \forall y{:}S_i \ \exists z{:}S_i \ (\mathrm{def}_i(x{:}S_W) \rightarrow \mathrm{def}_i[x{:}S_W \, y{:}S_i]) \wedge$
	$(\mathrm{def}_i(x{:}S_W) \rightarrow x{:}S_W \approx [x{:}S_W \, y{:}S_i \, z{:}S_i])$
4	$\Box_i p \rightarrow \Box_i \Box_i p$
	$\forall x{:}S_W \ \forall y, z{:}S_i \ \exists u{:}S_i \ (\mathrm{def}_i(x{:}S_W) \wedge \mathrm{def}_i[x{:}S_W \, y{:}S_i]) \rightarrow$
	$[x{:}S_W \, y{:}S_i \, z{:}S_i] \approx [x{:}S_W \, u{:}S_i]$
5	$\Diamond_i p \rightarrow \Box_i \Diamond_i p$
	$\forall x{:}S_W \ \forall y, z{:}S_i \ \exists u{:}S_i \ (\mathrm{def}_i(x{:}S_W) \rightarrow \mathrm{def}_i[x{:}S_W \, z{:}S_i]) \wedge$
	$(\mathrm{def}_i(x{:}S_W) \rightarrow [x{:}S_W \, y{:}S_i] \approx [x{:}S_W \, z{:}S_i \, u{:}S_i])$
G	$\Diamond_i \Box_i p \rightarrow \Box_i \Diamond_i p$
	$\forall x{:}S_W \ \forall y, z{:}S_i \ \exists u, u'{:}S_i \ (\mathrm{def}_i(x{:}S_W) \rightarrow (\mathrm{def}_i[x{:}S_W \, y{:}S_i] \wedge \mathrm{def}_i[x{:}S_W \, z{:}S_i] \wedge$
	$[x{:}S_W \, y{:}S_i \, u{:}S_i] \approx [x{:}S_W \, z{:}S_i \, u'{:}S_i]))$
alt_1	$\Diamond_i p \rightarrow \Box_i p$
	$\forall x{:}S_W \ \forall y, z{:}S_i \ (\mathrm{def}_i(x{:}S_W) \rightarrow [x{:}S_W \, y{:}S_i] \approx [x{:}S_W \, z{:}S_i])$

Fig. 6. Axiom schemas and functional correspondence properties

of natural numbers. We use $\overline{x}$ to denote a sequence of variables $x_1, \ldots, x_k$, and we denote by 'ϵ' and '.' the empty sequence and the concatenation operation on sequences, respectively. Then the *polyadic optimised functional translation* π'_{of} is given by the following.

$$\begin{aligned}
\pi'_{of}(p, \overline{x}, k, \sigma) &= \begin{cases} Q_{p,\epsilon} & \text{if } \sigma = \epsilon \text{ and } k = 0 \\ Q_{p,\sigma}(x_1, \ldots, x_k) & \text{otherwise} \end{cases} \\
\pi'_{of}(\neg\varphi, \overline{x}, k, \sigma) &= \neg\pi'_{of}(\varphi, \overline{x}, k, \sigma) \\
\pi'_{of}(\varphi \wedge \psi, \overline{x}, k, \sigma) &= \pi'_{of}(\varphi, \overline{x}, k, \sigma) \wedge \pi'_{of}(\psi, \overline{x}, k, \sigma) \\
\pi'_{of}(\Box_i\varphi, \overline{x}, k, \sigma) &= \forall x_{n+1} \, (\mathrm{def}_{i,\sigma}(\overline{x}) \rightarrow \pi'_{of}(\varphi, \overline{x}.x_{k+1}, k{+}1, \sigma.i)) \\
\pi'_{of}(\Diamond_i\varphi, \overline{x}, k, \sigma) &= \mathrm{def}_{i,\sigma}(\overline{x}) \wedge \pi'_{of}(\varphi, \overline{x}.x_{k+1}, k{+}1, \sigma.i)
\end{aligned}$$

In this definition the variable sequence in the argument position two of π'_{of} represents the world, and its path from the initial world, where the formula in argument position one is true. The translation of a modal formula φ is given by $\pi'_{of}(\varphi, \epsilon, 0, \epsilon)$. In the case of the modal logic KD, and for any modal logic where an accessibility relation R_i is serial, all occurrences of $\mathrm{def}_{i,\sigma}$ can be replaced by the logical constant $\top$. The following is an easy consequence of the previous theorem.

Theorem 8. *For any multi-modal logic L with K-modalities and D-modalities only, a modal formula φ is satisfiable in L iff $\exists\overline{x}\, \pi'_{of}(\varphi, \epsilon, 0, \epsilon)$ is first-order satisfiable.*

4.3 Mixed syntactic-semantic translations

A recent development is the *axiomatic translation* method which has a mixed syntactic-semantic flavour, because it is based on the standard semantic translation method, but instead of using correspondence properties it incorporates some or all of the additional modal axioms into the first-order translation [86, 87]. The method can be viewed as an almost semantic approach obtained by partially reducing the Hilbert-style syntactic encoding according to the definitions of the logical operators. The general motivation of this approach is to substitute the first-order theory expressed as correspondence properties by a theory with better computational and mathematical properties. Of course it must be a minimal requirement that satisfiability and unsatisfiability are preserved by the substitute theory.

It is necessary to define some extra notation before proceeding with a formal definition of the axiomatic translation mapping. Let $\mathrm{Sf}(\varphi)$ denote the set of all subformulae of φ. If ψ denotes a modal formula then let $\sim\psi$ denote the complement of ψ, i.e. $\sim\psi =^{\mathrm{def}} \phi$ if $\psi = \neg\phi$, and $\sim\psi =^{\mathrm{def}} \neg\psi$ otherwise. For any subformula ψ of φ, let the following represent the *definition* of Q_ψ.

$$\begin{aligned}\mathrm{Def}(\psi) = {} & \forall x\,(Q_\psi(x) \rightarrow \pi(\psi, x)) \\ & \wedge \forall x\,(Q_\psi(x) \rightarrow \neg Q_{\sim\psi}(x)) \wedge \forall x\,(Q_{\sim\psi}(x) \rightarrow \pi(\sim\psi, x))\end{aligned}$$

Q_ψ is a new predicate symbols uniquely associated with the modal formula ψ, and $\pi(\psi, x)$ is a first-order formula (with one free variable x) given by:

$$\begin{aligned}\pi(p, x) &= \top & \pi(\neg p, x) &= \neg Q_p(x) \\ \pi(\psi \wedge \phi, x) &= Q_\psi(x) \wedge Q_\phi(x) & \pi(\neg(\psi \wedge \phi), x) &= Q_{\sim\psi}(x) \vee Q_{\sim\phi}(x) \\ \pi(\Box_i\psi, x) &= \forall y\,(R_i(x, y) \rightarrow Q_\psi(y)) & \pi(\neg\Box_i\psi, x) &= \exists y\,(R_i(x, y) \wedge Q_{\sim\psi}(y))\end{aligned}$$

Theorem 9. *Let L be a first-order definable propositional multi-modal logic $K_{(m)}\varDelta$ which is sound and complete. For any modal formula φ, (i) φ is satisfiable in L iff $\mathrm{Corr}(\varDelta) \wedge \exists x\, Q_\varphi(x) \wedge \bigwedge\{\mathrm{Def}(\psi) \mid \psi \in \mathrm{Sf}(\varphi)\}$ is first-order satisfiable, and (ii) the translation can be computed in linear time.*

The definition of the encoding in this theorem is based on the standard relational semantics and structural transformation which introduces new symbols for each modal subformula. In the axiomatic translation the correspondence properties are replaced by a set of instances of so-called *schema clauses*. Figure 7 lists the schema clauses of some common axioms. They can be easily read off from the axioms or can be automatically computed from the modal axioms with the theorem prover MSPASS. The clauses are assumed to be closed under universal quantification of the free (first-order) variables. For each $\mathcal{A} \in \varDelta$, let $\mathfrak{X}_\mathcal{A}$ be a predetermined set of modal formulae. The intention is that each $\mathfrak{X}_\mathcal{A}$ is the instantiation set for the axiom $\mathcal{A}$. Let $\mathrm{Ax}^\mathcal{A}(\psi)$ be the conjunction of (the universal closure of) all clauses $C\{p/\psi\}$, where C is a schema clause in the schema clause set associated with $\mathcal{A}$. Further, let $X =^{\mathrm{def}} \{\mathrm{Ind}^\mathcal{A}(\psi) \mid \mathcal{A} \in \varDelta,\ \psi \in \mathfrak{X}_\mathcal{A}\}$, where

	Axiom $\mathcal{A}$	Schema clause $\mathrm{Ax}^{\mathcal{A}}(p)$
T	$\Box_i p \rightarrow p$	$\neg Q_{\Box_i p}(x) \vee Q_p(x)$
B	$\neg\Box_i\neg\Box_i p \rightarrow p$	$\neg R_i(x,y) \vee \neg Q_{\Box_i p}(y) \vee Q_p(x)$
D	$\Box_i p \rightarrow \neg\Box_i\neg p$	$\neg Q_{\Box_i p}(x) \vee Q_{\neg\Box_i\neg p}(x)$
4	$\Box_i p \rightarrow \Box_i\Box_i p$	$\neg Q_{\Box_i p}(x) \vee \neg R_i(x,y) \vee Q_{\Box_i p}(y)$
alt_1	$\neg\Box_i\neg p \rightarrow \Box_i p$	$\neg Q_{\neg\Box_i p}(x) \vee Q_{\Box_i\neg p}(x)$
5	$\neg\Box_i\neg\Box_i p \rightarrow \Box_i p$	$\neg R_i(x,y) \vee \neg Q_{\Box_i p}(y) \vee Q_{\Box_i p}(x)$

Fig. 7. Schema clauses for the axiomatic translation

$\mathrm{Ind}^{\mathcal{A}}(\psi)$ denotes the set of modal formulae occurring in the indices of the instances of the clauses associated with $\mathcal{A}$ as determined by the instance set $\mathfrak{X}_{\mathcal{A}}$. Then, if φ is a modal formula, the *axiomatic translation* of φ for $K_{(m)}\Delta$, relative to $\{\mathfrak{X}_{\mathcal{A}} \mid \mathcal{A} \in \Delta\}$, is the following conjunction.

$$\exists x\, Q_\varphi(x) \wedge \bigwedge\{\mathrm{Def}(\psi) \mid \psi \in \mathrm{Sf}(\varphi)\}$$
$$\bigwedge\{\mathrm{Ax}^{\mathcal{A}}(\psi) \mid (\mathcal{A}) \in \Delta,\ \psi \in \mathfrak{X}_{\mathcal{A}}\}$$
$$\bigwedge\{\mathrm{Def}(\psi) \mid \psi \in \mathrm{Sf}(X)\}$$
$$\bigwedge\{\forall x\, (\neg Q_{\sim\psi}(x) \rightarrow Q_\psi(x)) \mid \psi \in \mathrm{Sf}(X) \cup \mathrm{Sf}(\varphi)\}$$

There is an implicit restriction in this definition to axioms with one free variable, but the definition can be easily generalised for modal axioms in more than one variable.

In [86, 87] we have proved the following soundness and completeness results.

Theorem 10. *Let L be a consistent propositional modal logic $K_{(m)}\Delta$ with Δ a finite set. Let φ be any L-formula and assume $\bigcup\{\mathfrak{X}_{\mathcal{A}} \mid \mathcal{A} \in \Delta\}$ is a (finite) set of L-formulae. If φ is L-satisfiable then the axiomatic translation of φ for L relative to $\{\mathfrak{X}_{\mathcal{A}} \mid \mathcal{A} \in \Delta\}$ is first-order satisfiable.*

Theorem 11. *For each of the modal logics K4, KT, KD, KB, Kalt_1, KT4, KTB, KDB, KD4, K5, K4B, KT4B, or their fusions, and any modal formula φ, there is an effectively computable set $\mathfrak{X} =^{\mathrm{def}} \{\mathfrak{X}_{\mathcal{A}} \mid \mathcal{A} \in \Delta\}$ such that (i) φ is satisfiable in $K_{(m)}\Delta$ iff the axiomatic translation of φ relative to $\mathfrak{X}$ is first-order satisfiable, and (ii) the translation can be computed in linear time.*

In [87] we also show that the axiomatic translation can be applied to extensions of K with some generalised axioms ($alt_1^{\kappa_1,\kappa_2}$, 4^κ, 5^κ).

In addition, we show that the classical translation using correspondence properties and the axiomatic translation can be used together. We refer to this as the *combined axiomatic-relational translation*.

Theorem 12. *Let L be a consistent propositional modal logic $K_{(m)}\Delta$ with Δ a finite set. Suppose Δ can be partitioned into two sets, Δ' and Δ'', so that all axioms occurring in Δ' are first-order definable. Further suppose $\bigcup\{\mathfrak{X}_{\mathcal{A}} \mid \mathcal{A} \in \Delta''\}$*

is a finite set, and φ is any L-formula. If φ is L-satisfiable then the conjunction of $\mathrm{Corr}(\Delta')$ *and the axiomatic translation of φ relative to $\{\mathfrak{X}_{\mathcal{A}} \mid \mathcal{A} \in \Delta''\}$ is first-order satisfiable.*

This theorem gives a general soundness result for the combined axiomatic-relational translation method. Completeness can be proved for *S5* where the correspondence properties are used for the axioms T and B, while for the axiom *4* instantiations of the schema clause for *4* with all box subformulae of the input problem are sufficient. Other completeness results can be given for the logics *KDB* and *KD4*, and their fusions with any of the logics in Theorem 11. The axiomatic translation has the advantage that it can reduce also second-order modal logics to first-order logic. We expect it can be shown that any modal logic complete via the filtration construction can be reduced to first-order logic by this method. All the results are also true in the presence of non-logical axioms and for global satisfiability.

4.4 Other translations

There are many more ways of interpreting modal logics in other logics. Due to space restrictions we were forced to be selective about which translation methods to discuss. Other translations worth mentioning are the following. First, Areces et al [3] introduced a tree layered translation for the basic modal logic into first-order logic. It can be shown that there is a one-to-one correspondence between this translation and the (optimised) functional translation. Methods based on this translation can therefore be linearly simulated with the (optimised) functional translation method.

Second, with the translation of Demri and De Nivelle [22] all modal regular grammar logics with converse $RGL(^{\smile})$ can be reduced to GF^2, the two-variable guarded fragment, and decided by decision procedures for GF^2. This is a strong result because many of the common modal logics belong to $RGL(^{\smile})$. In some instances it can be easily seen that the axiomatic translation is equivalent to the translation of Demri-De Nivelle. In other instances the connection appears to be less immediate and deserves further investigation.

Third, translations to logics other than first-order logic exist as well. There is also a close relationship between modal logics and particularly the family of dynamic modal logics to description logics; in essence, modal logics can be viewed as notational variants of description logics and vice versa. Further, it is easy to see that many traditional modal logics can be embedded into suitably expressive dynamic modal logics, e.g. Peirce logic and $K_{(m)}(\neg)$ or $K_{(m)}(\top)$ (or even just $K_{(m)}$ possibly with serial modalities). In order to exploit fast SAT procedures we might even prefer to translate modal logics into propositional logic. The possibilities are endless.

5 Modal fragments, first-order fragments and clausal classes

Where do the different translations take us? Every translation described in the previous section associates a particular fragment of first-order logic to the modal source logic. Since the translations are all sound and complete it is immediate that any first-order logic theorem prover can be used as a prover for the considered source logic. Since all the translations are computable by linear functions the different target fragments inherit all the essential properties of the source modal logic, e.g. decidability, finite model property, computational complexity, etc. Our aim is to use existing first-order logic provers and we want decision procedures for the fragments of first-order logic which correspond to decidable modal logics. For this reason we take a closer look at the decidability of the target logics of the different translations and review the most important decidable clausal classes which are relevant.

5.1 Syntactic translation

Syntactic translations, i.e. encodings of Hilbert-style axiomatisations, take us to fragments of first-order logic with function symbols. Very few papers can be found in the literature on this kind of translation for modal logics. Consequently there are currently more open questions than answers. For example, since the 'syntactic modal fragments' are decidable for decidable modal logics, are there resolution methods for deciding these fragments, or can such methods be developed? Are solvable first-order fragments known which subsume these fragments?

5.2 Relational translation

First-order target logics. The target logic of the basic modal logic $K_{(m)}$ obtained via the classical relational translation is the *(relational) modal fragment* of first-order logic [92]. The modal fragment is the monadic fragment of first-order logic but the quantifiers are replaced by conditional quantifiers of the form $\forall y\, R(\cdot, y) \rightarrow \ldots$ and $\exists y\, R(\cdot, y) \wedge \ldots$ involving binary predicate symbols. A generalisation of this fragment is the *guarded fragment* and its many extensions [1, 2, 42, 41]. The guarded fragment has many of the same good properties as modal logics. It is decidable, it has the finite model property, it has the tree model property, etc. Thus, since its introduction the guarded fragment has taken over the role of the *two-variable fragment* of first-order logic (FO^2) as being regarded as a good generalisation of the modal fragment. FO^2 is decidable but Craig interpolation, Beth definability, and invariance for bisimulation fail. The guarded fragment subsumes more than just the (relational) modal fragment, it is also the target logic of a number of extensions of $K_{(m)}$, for example, it encompasses the dynamic modal logic $K_{(m)}(\wedge, \vee, ^{\smile}, id, ?)$ via the relational translation (cf. [87]). The standard translation of any formula with relational negation is not a guarded formula however.

FO^2, by contrast, has the advantage that it subsumes the relational fragment associated with dynamic modal logics with relational negation, for example Boolean modal logic, i.e. the logic $K_{(m)}(\neg, \wedge)$. It seems that the most expressive dynamic modal logic subsumed by $FO^2(\approx)$, the two-variable fragment with equality, is the logic $K_{(m)}(\neg, \wedge, {}^{\smile}, id, {}^{c})$.

There are other solvable first-order fragments which encompass modal logics via the relational translation method. One is the class $\overline{\mathrm{DK}}$ containing conjunctions of formulae in the dual of *Maslov's class K* [58]. This logic contains a variety of classical, solvable fragments, including the monadic class MON, the initially extended Skolem class $[\exists^*\forall\exists^*, \forall^*]$, and the Gödel class $\exists^*\forall^2\exists^*$. FO^2 (without equality) can also be embedded into $\overline{\mathrm{DK}}$ in a satisfiability-equivalence preserving way. Perhaps the most expressive dynamic modal logic subsumed by $\overline{\mathrm{DK}}$ is the extension $K_{(m)}(\neg, \wedge, {}^{\smile}, ;^{\mathsf{pos}}, {}^{c})$, where $;^{\mathsf{pos}}$ means that occurrences of composition are restricted to positive occurrences only.

Another solvable first-order fragment which encompasses modal logics via the relational translation is *fluted logic* (*FL*). Fluted logic was introduced by Quine as part of the definition of predicate functor logic [75, 76]. Fluted logic and extensions of fluted logic with binary converse and equality were shown to be decidable by Purdy [73, 74]. Fluted logic is actually quite closely related to modal logic, because the *dyadic fluted fragment*, i.e. the set of fluted formulae over unary and binary predicate symbols, is in fact the target logic of $K_{(m)}(\neg, \wedge)$ via the relational translation. One can prove [56] the following:

Theorem 13. *(i) For any formula φ in $K_{(m)}(\neg, \wedge)$, the formula $Qx\, \pi_r(\varphi, x)$, where $Q \in \{\forall, \exists\}$, is a dyadic fluted formula. (ii) For any closed dyadic fluted formula ψ there is a formula φ in $K_{(m)}(\neg, \wedge)$ such that ψ is logically equivalent to $Qx\, \pi_r(\varphi, x)$, where $Q \in \{\forall, \exists\}$.*

From a modal logic perspective, this result states that the dyadic fragment of fluted logic is the *relational modal fragment* of first-order logic associated with Boolean modal logic $K_{(m)}(\neg, \wedge)$. Analogous statements are true for $FL({}^{\smile})$, i.e. fluted logic with converse, respectively $FL({}^{\smile}, \approx)$, i.e. fluted logic with converse and equality, and $K_{(m)}(\neg, \wedge, {}^{\smile})$, respectively $K_{(m)}(\neg, \wedge, {}^{\smile}, id)$.

More frame correspondence properties belong to $FO^2(\approx)$, $\overline{\mathrm{DK}}$ and the fluted logic $FL({}^{\smile}, \approx)$ than the guarded fragments, but there are correspondence properties which do not belong to any of these fragments. Most notable examples are: transitivity and Euclideanness, the correspondence properties of the axioms *4* and *5*, respectively. Transitivity and Euclideanness are formulae which belong to the Bernays-Schönfinkel class, i.e. the $\exists^*\forall^*$ prefix class, which is decidable. Unfortunately, this is not of general use, because few modal formulae reduce to this class by the relational translation mapping; local satisfiability problems of modal formulae in which no existential modal operators occur below universal modal operators do. Here, by existential (universal) modal operators we mean positive (negative) occurrences of diamond operators and negative (positive) occurrences of box operators.

Solvable clausal classes. It is usually the case that the class of clauses corresponding to a solvable first-order fragment can be defined more generally. Two clausal clauses important in connection with modal logics are the class of guarded clauses and a class called DL^*.

The class of *guarded clauses* is a generalisation of the clausal class corresponding to the guarded fragment introduced by Ganzinger and De Nivelle [30]. The definition of this class makes use of the notions of shallow terms, simple literals and simple clauses, which are defined as follows. A term is *shallow* iff either it is a variable or a term $f(t_1, \ldots, t_n)$ such that each t_i is a variable or a constant ($0 \leq n$, $1 \leq i \leq n$). A literal L is *simple* iff each term in L is shallow, and a clause C is *simple* iff all literals in C are simple. A simple clause C is *guarded* iff it satisfies one of the following conditions:

1. C is a positive, non-functional, single-variable clause.
2. Every functional subterm of C contains all the variables of C, and, if C is non-ground, then C contains a non-functional negative literal, called a *guard*, which contains all the variables of C.

The class of all guarded clauses over the first-order language without equality is denoted by GC, with equality the class is denoted by $GC(\approx)$. The class $GC(\approx)$ is in fact slightly more general than the class of guarded formulae. In the next theorem the notation $^{c\,\mathrm{pos}}$ means that occurrences of cylindrification are restricted to positive occurrences.

Theorem 14. *The (global) satisfiability problem in $K_{(m)}(\wedge, \vee, ^{\smile}, ;^{\mathrm{pos}}, id, ?, ^{c\,\mathrm{pos}})$ with respect to a set of non-logical axioms of a formula φ is linearly reducible via the relational translation mapping to $GC(\approx)$.*

The class DL^* [20] is a variation of the class of DL-clauses, which was introduced in [53] with the purpose of handling expressive description logics. For reasons of simplicity we assume that all clauses are maximally split (i.e. the clauses cannot be partitioned into distinct variable-disjoint subclauses). A maximally split clause C is a *DL^*-clause* iff the following conditions are satisfied.

1. All literals are unary, or binary.
2. There is no nesting of function symbols.
3. Every functional term in C contains all the variables of C.
4. Every binary literal (even if it has no functional terms) contains all the variables of C.

Theorem 15. *The (global) satisfiability problem of a formula φ in the logic $K_{(m)}(\neg, \wedge, ^{\smile}, ;^{\mathsf{pos}}, ^{c})$ with respect to a set of non-logical axioms is linearly reducible via the relational translation mapping to a set of clauses in DL^*.*

It is still the case that the clausal form of some relational correspondence properties including transitivity and Euclideanness do not belong to either GC or DL^*. The clausal classes associated with fluted logic [84] or Maslov's class $\overline{\mathrm{DK}}$ [51] are no help here either.

5.3 Semi-functional translation

According to Theorem 6, a modal formula ϕ is $\mathsf{K}\mathit{\Delta}$-satisfiable iff $\pi_{sf}(\phi, \epsilon) \wedge \mathrm{SF}(\mathit{\Delta})$ is satisfiable, where $\mathrm{SF}(\mathit{\Delta})$ is a first-order formula representing the frame properties of $K\mathit{\Delta}$ according to Figure 5, where $\mathit{\Delta}$ is a subset of $\{\mathit{4}, \mathit{5}, B, D, T\}$. First, looking just at the translation mapping π_{sf}, we can see that the resulting first-order formulae do not belong to the guarded fragment, fluted logic, nor to the two-variable fragment due to the presence of the binary function symbol $[\_\,\_]$. For the same reason, the clauses corresponding to the semi-functional translation of a modal formula do not belong to the clausal classes associated with these three decidable fragments of first-order logic. However, the clauses belong the clausal class $\overline{\mathrm{DKC}}$ [51, 49] corresponding to the decidable class $\overline{\mathrm{DK}}$. Inspecting the formulae in Figure 5, one can see that the same is true for the clauses we obtain from the axioms D, T, B, and their combinations.

Theorem 16 ([49]). *Let $\mathit{\Delta}$ be any combination of the axiom schemas D, T, and B. Let φ be a modal formula and let N be the clausal form of $\pi_{sf}(\varphi, \epsilon) \wedge \mathrm{SF}(\mathit{\Delta})$. Then N belongs to the clausal class $\overline{\mathrm{DKC}}$.*

By contrast, the first-order formulae and clauses corresponding to the axiom schemas $\mathit{4}$ and $\mathit{5}$, and their combination with other axiom schemas do not belong to $\overline{\mathrm{DKC}}$. In [49] two decidable clausal classes are defined, the class of *small SF-clauses* and the class of *SF-clauses.*

Theorem 17 ([49]). *Let $\mathit{\Delta}$ be the axiom schema $\mathit{5}$, or its combination with $\mathit{4}$, D, and T. Let φ be a modal formula in negation normal form and N be the clausal form of $\pi_{sf}(\varphi, \epsilon) \wedge \mathrm{SF}(\mathit{\Delta})$. Then N consists only of small SF-clauses.*

Theorem 18 ([49]). *Let $\mathit{\Delta}$ be any combination of the axiom schemas $\mathit{4}$, D, and T. Let φ be a modal formula and N be the clausal form of $\pi_{sf}(\varphi, \epsilon) \wedge \mathrm{SF}(\mathit{\Delta})$. Then N consists only of SF-clauses.*

5.4 Optimised functional translation

The target logics of the optimised functional translation is a class of *path logics* of which the basic path logic is the weakest logic. Basic path logic corresponds to the basic modal logics which in this case are $K_{(m)}$ and $K(D)_{(m)}$, i.e. $K_{(m)}$ adjoined with serial modalities. Basic path logic is a fragment of the monadic Bernays-Schönfinkel class with one designated two-place function symbol and a constant. It is possible to prove that the monadic class with one binary function symbol is undecidable. Still, if we consider translations obtained via the polyadic optimised functional translation π'_{of}, then we can observe the following.

Theorem 19. *The polyadic optimised functional translation π'_{of} of any modal formula is equivalent to a formula in the Bernays-Schönfinkel class.*

In fact, the target logic of the π'_{of} variation of the optimised functional translation for $K_{(m)}$ (possibly adjoined with serial modalities) is a fragment of the Bernays-Schönfinkel class (and predicate symbols are not limited to monadic predicate symbols). Formulae in the π'_{of} fragment of $K(D)_{(m)}$ and formulae in path logics satisfy an interesting syntactic restriction called *prefix-stability.* Prefix stability says that every variable in $\pi_{of}(\varphi, x{:}S_W)$ (and $\pi'_{of}(\varphi, \epsilon, 0, \epsilon)$) has a unique prefix. This property is due to a characteristic ordering of variables in the path terms determined by the structure of modal formulae and is a reflection of the property of modal logic that the truth of a formula in a world depends only on the truth of subformula in the world and does not depend on predecessor worlds. Prefix stability is fundamental to the optimised functional translation, because it provides justification for the non-standard exchange of quantifiers (step (ii) in the transformation described on page 16). As a consequence the clausal classes corresponding to path logics have two important properties: (i) their input clauses do not contain any Skolem terms other than Skolem constants, and (ii) every occurrence of a variable or constant in an input clause has a unique prefix. These properties are preserved by inference steps in essentially any resolution procedure for those path logics for which the background theory does not include equational literals [81, 82].

Although we are focussing in this paper on the optimised functional translation there is an interesting observation worth making about the (non-optimised) functional translation. Suppose we take the variation of the functional translation, used in the definition of π'_{of}, where the paths are encoded by k-ary argument sequences rather than path terms. We refer to it here as the π'_f translation. It can be shown that the π'_f translation of any modal formula is a fluted formula [56, 81]. It therefore turns out that, besides dyadic fluted logic, there is another natural fragment of fluted logic relevant to modal logic. This is the *functional modal fragment*, a logic defined by Herzig in [45] with the original name 'ordered first-order logic'. This fragment is the target logic of the mentioned π'_f variation of the functional translation mapping for local satisfiability in $K(D)_{(m)}$. As a consequence all of the properties of $K(D)_{(m)}$ carry over to the functional modal fragment, in particular also the applicability of the non-standard quantifier exchange operator which is used in the optimised functional translation mapping. With this operator the functional modal fragment can be mapped into the Bernays-Schönfinkel class [50]. The transformed fragment coincides with the target logic of the optimised functional translation given by π'_{of}.

5.5 Axiomatic translation

A drawback of the guarded fragments when using the relational translation is that the correspondence properties such as transitivity, Euclideanness and functionality are not guarded formulae, and the guarded fragment extended with transitivity of binary predicates is undecidable [42] (but the monadic two-variable guarded fragment with transitive guards is decidable as shown by Ganzinger, Meyer, and Veanes [32]). Therefore a number of common modal logics apparently lie beyond the scope of the guarded fragment. Similar observations

	GF	$\exists^*\forall^*$	FL	$FL(^\smile)$	$\overline{\text{DK}}$	GC	DL^*	$\overline{\text{DKC}}$
$[\exists^*\forall\exists^*, \forall^*]$					●		●	●
$\exists^*\forall^2\exists^*$					●		●	●
FO^2				●	●		●	●
π_r fragment for $K_{(m)}$ (1)	●		●	●	●	●	●	●
π_r fragment for $K_{(m)}(\neg, \wedge)$ (2)			●	●	●		●	●
π'_f fragment for $K_{(m)}$ (3)			●	●				
π'_{of} fragment for $K_{(m)}$ (4)		●						
π_{sf} fragment for $K_{(m)}$								●

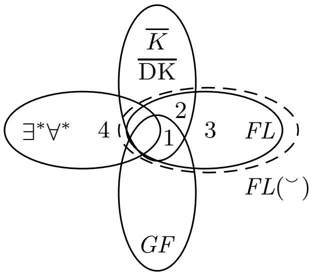

Fig. 8. Modal fragments, decidable first-order fragments and clausal classes

can be made for the other solvable fragments. Are important logics like *S4*, *S5*, *KD45* therefore outside the reach of decision procedures based on translation methods? The answer is no, the situation changes when using the axiomatic translation instead of the relational translation.

Theorem 20 ([87]). *(i) The axiomatic translation of any modal formula is equivalent to a GF^2 formula. (ii) The axiomatic translation of any modal formula can be linearly reduced to a set of DL^* clauses. (iii) The axiomatic translation of any modal formula can be linearly reduced to the modal fragment.*

Thus, the axiomatic translation always reduces (local and global satisfiability) problems (with non-logical axioms) in any of the traditional modal logics, for which completeness of the axiomatic translation can be proved, to GF^2 as well as DL^* and there are standard methods to reduce them to the modal fragment. This is the case for all the logics given in Theorem 11. The mixed axiomatic and relational translation, where some axioms are represented by correspondence properties while others are represented by an internalisation of axioms, might of course cross the boundaries of GF^2 and/or DL^*; this depends on whether the correspondence properties can be expressed in GF^2 (DL^*) or not.

5.6 Summary

We have seen that the basic modal logic $K_{(m)}$ can be embedded in one way or another into all the important decidable first-order fragments. This is summarised

	GF	FO^2	$\overline{\mathrm{DK}}$	FL	$FL({}^\smile)$	GC	DL^*	$\overline{\mathrm{DKC}}$
$K_{(m)}(\wedge, \vee, {}^\smile, {}^{c\ \mathrm{pos}})$	●	●	●	●	●	●	●	●
$K_{(m)}(\wedge, \vee, {}^\smile, ;^{\mathrm{pos}}, {}^{c\ \mathrm{pos}})$			●			●	●	●
$K_{(m)}(\wedge, \vee, {}^\smile, {}^{c})$		●	●		●		●	●
$K_{(m)}(\wedge, \vee, {}^\smile, ;^{\mathrm{pos}}, {}^{c})$			●				●	●
$K_{(m)}(\neg, \wedge)$		●	●	●	●		●	●
$K_{(m)}(\neg, \wedge, {}^\smile, {}^{c})$		●	●		●		●	●
$K_{(m)}(\neg, \wedge, {}^\smile, ;^{\mathrm{pos}}, {}^{c})$			●				●	●

Fig. 9. Dynamic modal logics and decidable fragments

	$GF(\approx)$	$FO^2(\approx)$	$FL({}^\smile, \approx)$	$GC(\approx)$
$K_{(m)}(\wedge, \vee, {}^\smile, id, ?)$	●	●	●	●
$K_{(m)}(\wedge, \vee, {}^\smile, ;^{\mathrm{pos}}, id, ?)$				●
$K_{(m)}(\wedge, \vee, {}^\smile, id, {}^{c})$		●	●	
$K_{(m)}(\neg, \wedge, {}^\smile, id, {}^{c})$		●	●	

Fig. 10. Dynamic modal logics and decidable fragments with equality

in Figure 8. A dot means that the logic associated with the row is subsumed by the logic associated with the column. Figures 9 and 10 summarise the relationship between dynamic modal logics and decidable first-order fragments with and without equality into which they can be embedded by the relational translation method.

6 Decision procedures

From the previous section a number of decidability results can be formulated for modal logics as consequences of the decidability of the first-order fragments into which they can be reduced. We can also use the decision procedures available for these fragments as decision procedures for the modal logics they incorporate. In this section we review results that have been obtained with different instances of the resolution framework.

Before we go on we want to note that most modern first-order logic theorem provers are implementations based on the Bachmair-Ganzinger framework of resolution. All of the ordering and selection-based refinements described in this section and subsequent sections are actually implemented in theorem provers such as (M)SPASS, VAMPIRE and *E*. Consequently, these provers immediately provide practical decision procedures for all the modal logics and solvable fragments covered by the decidability results stated in this section. Since the majority of the refinements are specified in general ways and can be realised in a number of ways also outside the framework of Bachmair-Ganzinger, it is possible to use resolution provers as decision procedures as well. Sometimes some modest implementation effort is necessary for implementing the required translation mapping, but many of translation mappings are already implemented in the theorem prover MSPASS. MSPASS [54, 83] serves as the experimental platform for much of our research. It is an extension of SPASS and differs from SPASS in that it accepts also formulae as input from (traditional and dynamic) modal logics, description logic and relational logic. We have also extended the converter of first-order logic formulae to clausal form. MSPASS supports non-logical axioms and global satisfiability/validity. In addition, it is possible to specify additional frame properties, or other first-order restrictions on the translated formulae. A converter to TPTP syntax is also available so that other first-order theorem provers can be used.

6.1 Unrefined resolution

First we consider whether decision procedures can be based on unrefined resolution, that is, there is no ordering restriction and the selection function S is empty.

Using the optimised functional translation. The basic unrefined resolution calculus R plus condensing provides a decision procedure for propositional logic, but it does not decide the standard (relational) modal fragment of the basic modal logic [81, 82]. The latter may appear surprising especially since we know that almost any other proof methods (tableaux, sequent systems, natural deduction) decide the modal logic $K_{(m)}$. However, on the one hand, modal tableau decision procedures can be regarded as *refinements* of resolution which are decision procedures for $K_{(m)}$ (see Section 7). On the other hand, it has been shown that the basic modal logics *can* be decided by unrefined resolution when the optimised functional translation, or essentially equivalent translations, are used.

Theorem 21 ([81, 82]). *R with condensing is a decision procedure for (i) the clausal form of the basic path logic, and (ii) the clausal form of the $\pi_{of'}$ fragment for $K(D)_{(m)}$.*

This result can be formulated as giving general criteria under which unrefined resolution with condensing, theory unification and theory normalisation provides a decision procedure for path logics extended with equational theories and consequences for modal logics that meet these conditions; see [81, 82] for details. These results are interesting both from a theoretical and practical perspective because Theorem 21 says that no special refinements are required and, in fact:

Theorem 22 ([81, 82]). Any refinement *of resolution with condensing is a decision procedure for basic path clauses and the local satisfiability of problems in $K(D)_{(m)}$.*

Corollary 3. *For any admissible ordering and any selection function any resolution procedure based on R^{red} or $\mathsf{R}^{\text{red}}_{\text{sp}}$ with condensing are decision procedures for basic path clauses and local satisfiability problems in $K(D)_{(m)}$.*

Since condensing is explicitly present in many resolution provers, or if not, can be simulated by factoring and subsumption deletion, which are present in all state-of-the-art resolution provers, Theorem 22 means that *any complete and fair* use of a first-order resolution prover is a decision procedure for $K(D)_{(m)}$, when the (monadic or polyadic) optimised functional translation is used. This gives essentially complete freedom in the way the theorem provers can be used for these logics without sacrificing guaranteed (normal) termination. In other words, unrestricted use and fine-tuning of ordering restrictions, selection functions and redundancy criteria is possible. This implies that e.g. (M)SPASS, VAMPIRE and E are decision procedures for the translation of $K(D)_{(m)}$. What about tableaux-based theorem provers for first-order logic? Since basic path logic is a fragment of the Bernays-Schönfinkel class and this class can be decided by almost all tableaux provers, it follows that these tableaux-based theorem provers (and instantiation-based methods, e.g. DCTP [90], first-order DPLL [10], the model evolution approach [13]) are also decision procedures for these kinds of problems.

6.2 Ordered resolution

Using the semi-functional translation. According to Theorem 17 the clausal form of the semi-functional translation of a formula φ in *K5* and its extensions by an arbitrary combination of the axiom schemas *4*, D, and T is a set of small SF-clauses. The satisfiability of a set of small SF-clauses can be decided by ordered resolution based on a general class of atom orderings [49]. Namely, let $\succ_{\mathcal{COV}}$ be any atom ordering compatible with the multiset extension $\succ^s_{mul}$ of the strict subterm ordering $\succ^s$ on the complexity measure c_L on literals, where for any literal L, c_L is the multiset of arguments of L.

Theorem 23 ([49]). *Let Δ be a combination of the axiom schema 4, D, and T plus the axiom schema 5. Let φ be a modal formula in negation normal form and let N be the clausal form of $\pi_{sf}(\varphi, \epsilon) \wedge \mathrm{SF}(\Delta)$. Any derivation from N by ordered resolution and ordered factoring based on the ordering $\succ_{\mathcal{COV}}$ terminates.*

In analogy, according to Theorem 18 the clausal form of the semi-functional translation of a formula φ in K extended by an arbitrary combination of the axiom schemas *4*, D, and T is a set of SF-clauses. In contrast to the class of small SF-clauses, we need a selection function in addition to an atom ordering to ensure termination of derivations from sets of SF-clauses. We also need be more restrictive concerning the atom ordering that we are allowed to use.

Let $\succ$ be any total reduction ordering on ground terms in which the constant ϵ is the minimal term. For every ground literal L, let $c'_L = (\max_L, \mathrm{ar}_L, \mathrm{pol}_L, s_L)$ where (i) $\max_L$ is the maximal argument of L with respect to $\succ$, (ii) ar_L is the arity of L, (iii) pol_L is 1, if L is negative, and 0 otherwise, and (iv) s_L is 1, if L is a dyadic literal $\neg r(s, t)$ and $s \succ t$, and 0 otherwise, The ordering $\succ_c$ over the complexity measure is then the lexicographic combination of $\succ$, $>_{\mathbb{N}}$, $>_{\mathbb{N}}$, and $>_{\mathbb{N}}$. For example, if $s \succ t$, then the complexity of $r(s, t)$ is $(s, 2, 0, 1)$, whereas the complexity of $\neg r(t, s)$ is $(s, 2, 1, 0)$. Observe that the maximal term is the main criterion, and a negative literal is considered more complex than a positive literal with the same maximal term. Note that $\succ_c$ represents a strict partial and well-founded ordering on ground literals. Any total and well-founded extension (again denoted by $\succ$) of $\succ_c$ is an admissible ordering in the sense of [5]. Let $\succ_{\mathcal{ML}}$ be any such ordering. The ordering $\succ_{\mathcal{ML}}$ is lifted to non-ground expression in the standard manner.

In addition, we make use of a selection function $S_{\mathcal{ML}}$ which is defined as follows. If a ground clause C contains a negative dyadic literal of the form $\neg r(s, t)$ such that s is an occurrence of a $\succ$-maximal term in C, then S selects one such literal. No other literals are selected by $S_{\mathcal{ML}}$. A literal L is *selected* in a non-ground clause C, $L\sigma$ is selected in $C\sigma$, for all ground instances, by a substitution σ, of an inference with $C\sigma$ by ordered resolution or ordered factoring such that the ordering constraints are satisfied.

Theorem 24 ([49]). *Let Δ be any combination of the axiom schemas 4, D, and T. Let φ be a modal formula in negation normal form and let N be the clausal form of $\pi_{sf}(\varphi, \epsilon) \wedge \mathrm{SF}(\Delta)$. Any derivation from N by ordered resolution and ordered factoring with selection based on the ordering $\succ_{\mathcal{ML}}$ and the selection function $S_{\mathcal{ML}}$ terminates.*

Using the relational translation. We have seen that using the relational translation method many of the modal logics we consider reduce to the two decidable clausal classes, $GC(\approx)$ and DL^*.

Let $>_d$ be an ordering on terms defined as follows: $s >_d t$ if s is deeper than t, and every variable that occurs in t, occurs deeper in s. This ordering is lifted to atoms using a multiset extension. Let S be the selection function which selects all guard literals in each non-ground guarded clause. Let $\mathsf{R}^{\succ,S}$ denote any resolution procedure with condensing based on an ordering $\succ$ compatible with $>_d$, and the selection function S. In $\mathsf{S}^{\succ,S}$ the rules are those of $\mathsf{R}^{\succ,S}$ and for equality, ordered paramodulation is added.

Theorem 25 (Ganzinger and De Nivelle [30]). *(i) $\mathsf{R}^{\succ,S}$ is a decision procedure for GC. (ii) $\mathsf{S}^{\succ,S}$ is a decision procedure for $GC(\approx)$.*

Corollary 4. *Let L be a dynamic modal logic in the sublattice bounded by $K_{(m)}$ and $K_{(m)}(\wedge, \vee, {}^{\smile}, ;^{\mathrm{pos}}, id, ?, {}^{c\ \mathrm{pos}})$, let φ be an L-formula, and let Σ be a finite set of relational properties expressible in $GC(\approx)$. Then, (i) there is a linear transformation of $\Sigma \wedge \exists x\, \pi_r(\varphi, x)$ into a set N of $GC(\approx)$ clauses, (ii) any derivation from N in $\mathsf{S}^{\succ,S}$ terminates, and (iii) φ is unsatisfiable in L iff there is a refutation of N in $\mathsf{S}^{\succ,S}$. (Analogously, for global satisfiability and non-logical axioms.)*

The clausal class DL^* can be decided by standard ordering refinements of resolution based on the ordering $>_d$, without the need for a specific selection function. In particular, let $\mathsf{R}^{\succ}$ denote any resolution procedure with condensing based on an ordering $\succ$ compatible with $>_d$, and the empty selection function.

	Axiom	Correspondence property
Det	$[\beta]p \rightarrow \langle\gamma\rangle p$	$\forall x \exists y\,(R_\beta(x,y) \wedge R_\gamma(x,y))$
Sym	$\langle\alpha\rangle[\beta]p \rightarrow p$	$\forall x \forall y\,(R_\alpha(x,y) \rightarrow R_\beta(y,x))$
Gr	$[\beta]p \rightarrow [\alpha]p$	$\forall x \forall y\,(R_\alpha(x,y) \rightarrow R_\beta(x,y))$
Conf	$\langle\alpha\rangle[\beta]p \rightarrow \langle\gamma\rangle p$	$\forall x \forall y\,(R_\alpha(x,y) \rightarrow \exists z\,(R_\gamma(x,z) \wedge R_\beta(y,z)))$

Fig. 11. Modal axioms and their relational correspondence properties

Theorem 26 ([20, 53]). *$\mathsf{R}^{\succ}$ is a decision procedure for DL*.*

This theorem has the following consequences for modal logics [85].

Corollary 5. *Let L be a dynamic modal logic in the sublattice bounded by $K_{(m)}$ and $K_{(m)}(\neg, \wedge, ^{\smile}, ;^{\mathsf{pos}}, ^{c})$, and let φ be an L-formula, let Σ be a finite set of relational properties expressible in DL*. Then, (i) there is a linear transformation of $\Sigma \wedge \exists x\, \pi_r(\varphi, x)$ into a set N of DL* clauses, (ii) any derivation from N in $\mathsf{R}^{\succ}$ terminates, and (iii) φ is unsatisfiable in L iff there is a refutation of N in $\mathsf{R}^{\succ}$. (Analogously, for global satisfiability and non-logical axioms.)*

Corollary 6. *Let Δ be any finite set of instances of formulae in Figure 11, where α is a relational formula built from relational variables and disjunction only, while β and γ denote either a relational variable or a relational formula built from relational variables using disjunction and composition. Then, (i) the satisfiability problem in $K_{(m)}\Delta$ is decidable, and (ii) it can be decided by reduction to DL* (via the relational translation) and $\mathsf{R}^{\succ}$. (Analogously, for global satisfiability and non-logical axioms.)*

The above results are not stated in the most general form possible, because in fact a slight extension of *DL** can be decided by resolution (cf. [85]). However Corollary 5 cannot be strengthened further by removing the restriction on occurrences of composition.

Theorem 27. *The satisfiability problem in $K_{(m)}(\neg, \wedge, ;)$ and every logic in the family of dynamic modal logics extending $K_{(m)}(\neg, \wedge, ;)$ is undecidable.*

Using the axiomatic translation. Some of the common traditional modal logics which escape embedding into decidable fragments by the other translation methods can be decided when using the axiomatic translation method. The results stated here have been formulated and proved for local satisfiability without any non-logical axioms in [87], but are also true for global satisfiability and non-logical axioms.

Theorem 28 ([87]). *Let L be a (sound and) complete modal logic $K_{(m)}\Delta$. Then, L is decidable, whenever the following conditions are satisfied. (i) Δ is finite. (ii) For any L-formula φ, there are effectively computable sets $\mathfrak{X}_{\mathcal{A}}$ for each $\mathcal{A} \in \Delta$ such that, if the axiomatic translation of φ for L relative to $\{\mathfrak{X}_{\mathcal{A}} \mid \mathcal{A} \in \Delta\}$ is first-order satisfiable, then φ is satisfiable in L.*

Corollary 7 ([87]). *Each of the modal logics $K4$, KT, KD, KB, $K\mathit{alt}_1$, $KT4$, KTB, KDB, $KD4$, $K5$, $K4B$, $KT4B$, and their fusions are decidable.*

The decidability of the logics $K4^{\kappa}$, $K5^{\kappa}$ and $K\mathit{alt}_1^{\kappa_1,\kappa_2}$ can also be shown using the axiomatic translation method.

Theorem 29. *Both $\mathsf{R}^{\succ}$ and $\mathsf{R}^{\succ,S}$ decide the axiomatic translation of satisfiability problems in all modal logics satisfying the conditions of Theorem 28.*

Theorem 30 ([87]). *Let L be a (sound and) complete propositional modal logic $K_{(m)}\Delta$ and suppose Δ is partitioned into two sets Δ' and Δ''. Then, L is decidable, whenever the following conditions are satisfied. (i) Δ is finite. (ii) For each axiom $\mathcal{A}$ in Δ', $\mathrm{Corr}(\mathcal{A})$ is expressible in DL^*. (iii) For any L-formula φ, there are effectively computable sets $\mathfrak{X}_{\mathcal{A}}$ for each axiom $\mathcal{A}$ in Δ'' such that, if ψ is the axiomatic translation of φ relative to $\{\mathfrak{X}_{\mathcal{A}} \mid \mathcal{A} \in \Delta''\}$ then $\mathrm{Corr}(\Delta') \wedge \psi$ is satisfiable in first-order logic then φ is satisfiable in L.*

Theorem 31 ([87]). *$\mathsf{R}^{\succ}$ decides the encoding of satisfiability problems in all modal logics satisfying the conditions of Theorem 30.*

For example, $\mathsf{R}^{\succ}$ decides the combined axiomatic-relational translation of problems in the logics $KT4$, KDB and $KD4$.

The ideas and principles underlying the axiomatic translation are quite general and are applicable not just to traditional modal logics. For example, the principle can be used to obtain decision procedures for the monadic two-variable guarded fragment with transitivity studied in Ganzinger, Meyer and Veanes [32].

6.3 Selection-based resolution

The logic $K_{(m)}(\wedge, \vee, ^{\smile}, ^{c\,\mathrm{pos}})$, and dynamic modal logics below it, have the property that they can be decided by a refinement of resolution which is defined solely by a selection function of negative literals, and as a particular case hyperresolution. The results stated here are (in some cases slightly strengthened results) from [20, 53]. Let $\mathsf{H}^{\mathrm{taut}}_{\mathrm{sp}}$ denote the (unordered) hyperresolution calculus $\mathsf{H}^{\mathrm{red}}_{\mathrm{sp}}$ with splitting and at least tautology deletion. All results hold also for positive resolution and ordered hyperresolution $\mathsf{OH}^{\mathrm{red}}_{\mathrm{sp}}$ (the latter is more restrictive and means that the search space is smaller).

Theorem 32. *Let L be a dynamic modal logic in the sublattice bounded by $K_{(m)}$ and $K_{(m)}(\wedge, \vee, ^{\smile}, ^{c\,\mathrm{pos}})$. Let φ be any L-formula. Then, φ can be reduced via the relational translation into a clause set N such that, (i) any $\mathsf{H}^{\mathrm{taut}}_{\mathrm{sp}}$-derivation from N terminates, and (ii) φ is unsatisfiable in L iff there is a refutation of N by $\mathsf{H}^{\mathrm{taut}}_{\mathrm{sp}}$.*

Theorem 33. *Let L be a dynamic modal logic in the sublattice bounded by $K_{(m)}$ and $K_{(m)}(\wedge, \vee, ^{\smile}, ^{c\,\mathrm{pos}})$. Let Σ be a finite $\mathsf{H}^{\mathrm{taut}}_{\mathrm{sp}}$-saturated set of clauses consisting of two kinds of split components.*

1. *Clauses with at most two free variables, which are built from finitely many binary predicate symbols R_j, no function symbols, and containing at least one guard literal (that is, this literal is negative and includes all the variables of the clause).*
2. *Clauses built from one variable, finitely many function symbols (including constants), and finitely many binary predicate symbols R_j, with the restriction that (a) the argument multisets of all non-ground literals coincide, and (b) each literal which contains a constant is ground.*

Suppose φ is an L-formula. Then, φ can be reduced via the relational translation into a clause set N such that, (i) any $\mathit{H}^{\mathrm{taut}}_{\mathrm{sp}}$-derivation from $N \cup \Sigma$ terminates, and (ii) φ is unsatisfiable in L with respect to Σ iff there is a refutation of $N \cup \Sigma$ by $\mathit{H}^{\mathrm{taut}}_{\mathrm{sp}}$.

Ordered resolution and hyperresolution are refinements of resolution which traverse the search space in different manners. Ordering refinements are designed to prevent term depth growth, whereas in general in hyperresolution derivations terms get larger for translated modal formulae. An important technical problem is therefore to find criteria and mechanisms for preventing unbounded growth of terms. Our studies in [37, 38] have shown that, in general, terms do not grow indefinitely and hyperresolution procedures terminate when the clauses satisfy a certain acyclic dependency relation. This acyclicity condition is satisfied for the logics for which positive results are stated above. In the absence of this property hyperresolution ($\mathit{H}^{\mathrm{taut}}_{\mathrm{sp}}$ or $\mathit{OH}^{\mathrm{red}}_{\mathrm{sp}}$) is in general *not* a decision procedure for (the relational translation of) modal logics. What would be required is an additional mechanism for detecting repetitions. In many cases the use of a blocking deduction rule [52] suffices. At certain points during a derivation this rule adds equations $t_1 \approx t_2$ between ground terms t_1 and t_2 to the clause set, rendering inferences on literals involving the greater of the two terms redundant. One of the interesting properties of this approach is that completeness follows immediately from the general completeness result of Bachmair, Ganzinger, and Waldmann [8], only soundness needs to be established. The methods introduced in [12] provide an alternative solution.

6.4 Categorisation of relational properties

In Figure 12, we give a non-exhaustive categorisation of the expressibility of some relational frame properties in the mentioned classes to give the reader a glimpse into the scope of the results stated in this section.

7 Relationship to tableaux

It is possible to show that many forms of modal tableau calculi can be linearly simulated with standard techniques in resolution-based theorem proving. For example, in [55] we give details of how to simulate the single-step prefixed tableau

Relational property	GC	DL^*	Thm 33
$\forall x \forall y\, (R_\alpha(x,y) \rightarrow R_\beta(x,y))$	●	●	●
$\forall x \forall y\, (R_\alpha(x,y) \rightarrow R_\gamma(x,y))$	●	●	
$\forall x \exists y\, R_\beta(x,y)$		●	●
$\forall x \exists y\, R_\delta(x,y)$		●	
$\exists x \exists y\, R_\delta(x,y)$	●	●	
$\exists x \exists y\, R_\epsilon(x,y)$	●	●	●

α	is built from	relational variables and $\wedge,\ \vee,\ {}^{\smile}$
β	is built from	relational variables and $\wedge,\ \vee,\ {}^{\smile},\ \bot$
γ	is built from	relational variables and $\wedge,\ \vee,\ {}^{\smile},\ ;,\ {}^{c},\ \upharpoonleft,\ \downharpoonright,\ \bot$
δ	is built from	relational variables and $\wedge,\ \vee,\ {}^{\smile},\ ;,\ {}^{c},\ \upharpoonleft,\ \downharpoonright$ or $\neg,\ \wedge,\ \vee,\ {}^{\smile},\ {}^{c},\ \upharpoonleft,\ \downharpoonright$
ϵ	is built from	relational variables and $\wedge,\ \vee,\ {}^{\smile},\ ;$

Fig. 12. Categorisation of some relational properties

calculi of [59] using the hyperresolution calculus $H_{\mathrm{sp}}^{\mathrm{taut}}$ (see also [20, 46, 85]). Simulation results of tableau procedures for description logics can be found in [52, 53], see also [28].

From the proofs of these simulation results the view emerges that labelled semantic tableau algorithms are just hyperresolution with lazy translation to first-order logic. The same can be said for many tableau algorithms without the use of labels and without explicit representation of accessibility, but this is less obvious, cf. [46]. The following question arises as a consequence: Is it possible to develop new tableau inference calculi within the resolution framework? The answer is: yes. In [20] we show how the relationship between hyperresolution and tableau can be exploited for systematically developing sound, complete and terminating tableau procedures for dynamic modal logics. The tableau calculus extracted for $K_{(m)}(\wedge, \vee, {}^{\smile})$ is defined in Figure 13 (the formulae are w.lo.g. assumed to be in negation normal form). It follows from Theorem 32 that:

Theorem 34 ([20]). *The tableau calculus defined in Figure 13 is sound, complete and terminating for local satisfiability in $K_{(m)}(\wedge, \vee, {}^{\smile})$.*

The calculus in Figure 13 is unusual in that it requires the rules for the relational operators to be applied in two directions. Thus, the calculus comprises both elimination rules and introduction rules. Consequently the calculus can also be viewed as a restricted form of natural deduction calculus. While introduction rules could jeopardise decidability, the side conditions specify restrictions which imply that any procedure based on this calculus is a decision procedure. This means in particular that blocking mechanisms are required. To our knowledge the calculus in Figure 13 was the first non-resolution calculus to be defined for local satisfiability in $K_{(m)}(\wedge, \vee, {}^{\smile})$. Standard methods can be used to extend this calculus for testing global satisfiability and handling non-logical axioms.

$$(\bot)\ \frac{s:\psi,\ s:\neg\psi}{s:\bot} \qquad (\wedge)\ \frac{s:\psi\wedge\phi}{s:\psi,\ s:\phi} \qquad (\vee)\ \frac{s:\psi\vee\phi}{s:\psi \quad \mid \quad s:\phi}$$

$$(\Diamond)\ \frac{s:\langle\alpha\rangle\psi}{(s,t):\alpha,\ t:\psi}\ \text{ with } t \text{ new to the branch} \qquad (\Box)\ \frac{(s,t):\alpha,\ s:[\alpha]\psi}{t:\psi}$$

$$(\smile)\ \frac{(s,t):\alpha^{\smile}}{(t,s):\alpha} \qquad (\wedge^r)\ \frac{(s,t):\alpha\wedge\beta}{(s,t):\alpha,\ (s,t):\beta} \qquad (\vee^r)\ \frac{(s,t):\alpha\vee\beta}{(s,t):\alpha \quad \mid \quad (s,t):\beta}$$

$$(\smile_I)\ \frac{(t,s):\alpha}{(s,t):\alpha^{\smile}} \qquad (\wedge^r_I)\ \frac{(s,t):\alpha,\ (s,t):\beta}{(s,t):\alpha\wedge\beta} \qquad (\vee^r_I)\ \frac{(s,t):\alpha}{(s,t):\alpha\vee\beta}$$

Fig. 13. Tableau calculus for $K_{(m)}(\wedge,\vee,^{\smile})$. For the rules $(\smile_I)$, $(\wedge^r_I)$ and $(\vee^r_I)$ the side conditions are that the relational formulae in the denominator, i.e. $\alpha^{\smile}$, $\alpha\wedge\beta$ or $\alpha\vee\beta$, occur as subformulae of the relational formula γ of a box formula $s:[\gamma]\psi$ on the current branch. It is assumed that in a derivation no rule is applied twice to the same instance of the nominator.

It is actually easy to derive sound and complete tableau-like calculi via any of the translation methods we have discussed. When $\mathit{H}^{\mathrm{taut}}_{\mathrm{sp}}$ is a decision procedure for the appropriate clause forms then decidability follows also immediately.

Let us take the axiomatic translation. Figure 14 gives examples of tableau calculi which can be more or less immediately read off from the combination of the axiomatic translation and hyperresolution [85, 87]. It follows from Corollary 7 that:

Theorem 35 ([87]). *The tableau calculi defined in Figure 14 are sound and complete for local satisfiability in the logics specified in the figure.*

Similarly sound and complete tableau calculi can be derived for the logics: $K4^{\kappa}$, $Kalt_1^{\kappa_1,\kappa_2}$, $K5^{\kappa}$, $K4B$ and $KT4B$. Again, standard methods can be used to extend these calculi for testing global satisfiability and handling non-logical axioms.

Interestingly the calculi in Figure 14 either coincide with or are equivalent to labelled semantic tableau calculi developed in, e.g., [18, 40, 59]. This means that our implementation of the axiomatic translation and any first-order logic theorem prover supporting hyperresolution can be used as a tableau prover for these logics. Moreover, MSPASS can essentially be used as a tableau theorem prover for $K_{(m)}(\wedge,\vee,^{\smile})$ by just choosing the appropriate combination of flag settings that simulate the tableau inference steps for $K_{(m)}(\wedge,\vee,^{\smile})$. This shows that in resolution, not only do we have a theoretical framework for studying and developing tableau methods, but also that we can easily get fully operational tableau provers for free.

8 Automated model generation

A problem closely related to satisfiability problems is the problem of generating (counter-)models. Hyperresolution can be used for constructing Herbrand models

$$(\bot)\ \frac{s:\psi,\ s:\neg\psi}{s:\bot} \qquad (\neg)\ \frac{s:\neg\neg\psi}{s:\psi} \qquad (\wedge)\ \frac{s:\psi\wedge\phi}{s:\psi,\ s:\phi} \qquad (\neg\wedge)\ \frac{s:\neg(\psi\wedge\phi)}{s:{\sim}\psi \ \mid\ s:{\sim}\phi}$$

$$(\neg\Box)\ \frac{s:\neg\Box\psi}{(s,t):R,\ t:{\sim}\psi}\ \text{where } t \text{ is new to the branch} \qquad (\Box)\ \frac{(s,t):R,\ s:\Box\psi}{t:\psi}$$

$$(T)\ \frac{s:\Box\psi}{s:\psi} \qquad (D)\ \frac{s:\Box\psi}{s:\neg\Box\neg\psi} \qquad (B)\ \frac{(s,t):R,\ t:\Box\psi}{s:\psi}$$

$$(4)\ \frac{s:\Box\psi,\ (s,t):R}{t:\Box\psi} \qquad (alt_1)\ \frac{s:\neg\Box\psi}{s:\Box{\sim}\psi}$$

K :	(K)	KD :	$(K),(D)$	KT :	$(K),(T)$
KB :	$(K),(B)$	$K4$:	$(K),(4)$	$K alt_1$:	$(K),(alt_1)$
KTB :	$(K),(T),(B)$	KDB :	$(K),(D),(B)$	$KD4$:	$(K),(D),(4)$
$S4$:	$(K),(T),(4)$				

Fig. 14. Tableau calculi obtained via the axiomatic translation. It is assumed that in a derivation no rule is applied twice to the same instance of the nominator.

(and representations of Herbrand models) [28]. A *Herbrand interpretation* is a set of ground atoms. By definition a ground atom A is *true* in an interpretation H iff $A \in H$ and it is *false* in H iff $A \notin H$. The definition extends as expected to (non-)ground clauses, and sets of clauses.

The results below are consequences of properties of classes of range restricted clause sets. For range restricted clause sets hyperresolution procedures implicitly generate Herbrand models [17, 36, 38]. For a class of solvable range restricted clauses, if hyperresolution terminates on a clause set N without having produced a refutation then a model can be extracted from any complete, open branch in the derivation. The model is given by the set of ground unit clauses in the limit of the branch. In general the generated Herbrand model can be infinite, but using Theorems 32 and 33 we get (cf. [20, 55]):

Theorem 36. *The combination of the relational translation and $H^{\mathrm{taut}}_{\mathrm{sp}}$ can be used as a Herbrand model generator for the local satisfiability in any dynamic modal logic in the ideal generated by $K_{(m)}(\wedge,\vee,^{\smile})$.*

Theorem 37. *Let L and Σ be as in Theorem 33. For any modal formula locally satisfiable in L with respect to Σ a finite modal model can be effectively constructed on the basis of hyperresolution and the relational translation.*

This implies the finite model property for the relevant modal logics. By the simulation results in the previous section these results extend also to the corresponding tableau calculi.

Corollary 8 ([20]). *If L is a dynamic modal logic in the ideal generated by $K_{(m)}(\wedge,\vee,^{\smile})$, and φ is locally satisfiable in L then a finite modal model can* *be effectively constructed on the basis of the tableau calculus for L given by the appropriate subset of inference rules in Figure 13.*

With the exception of non-serial and non-functional modal logics it can be inferred from the completeness proofs in [87] that the model constructed by hyperresolution (with a blocking deduction rule) for the axiomatic encoding of a modal formula φ is a tree model. This model can be easily mapped into a corresponding modal (tree) model. In general this model is however not a model of the considered formula φ, but can be viewed as a skeleton model which can be completed with respect to the appropriate correspondence properties to a model of φ. Thus:

Theorem 38 ([87]). *Let L be any of following logics or their fusions: K4, KT, KB, KT4, KTB, , K5, K4B, KT4B. If a modal formula φ is L-satisfiable then (i) it is satisfiable in a model with at most $2^{\mathcal{O}(|\varphi|)}$ elements, i.e. L has the small model property, and (ii) resolution can be used to construct an L-model for φ.*

The result holds also for the logics $K4^\kappa$ and $K5^\kappa$.

In general Herbrand models are not unique and can be large. Methods for generating 'small' Herbrand models are therefore of interest. Various approaches to generating minimal Herbrand models with hyperresolution are known [11, 17, 43, 61]. An interpretation H is a *minimal Herbrand model* for a set N of clauses iff H is a Herbrand model of N and for no Herbrand model H' of N, $H' \subset H$ holds. With a moderate extension of hyperresolution $\mathit{H}_{\mathrm{sp}}^{\mathrm{taut}}$ it is possible to guarantee the generation of all and only minimal Herbrand models for any modal and description logics reducible to a decidable class of range restricted clauses. This follows from [17] and recent investigations of a fragment of the guarded fragment and the class $\mathcal{B}U$ [36, 37]. An alternative approach proposed in [36, 37] uses a variant of a local minimality test developed for propositional logic.

It is not difficult to see that model generation procedures and the mentioned minimal Herbrand model generation procedures can be developed by using hyperresolution, or ordered resolution with maximal selection, and any of the other translation methods. Because of the close connection to tableau, corresponding tableau procedures can be defined and all results carry over to this setting.

9 Automated correspondence theory

Modal correspondence theory is concerned with the study of the characteristic properties of Kripke frames in the semantics of traditional modal logics, see [92] and also e.g. [14, 57]. One of the most general results in correspondence theory is Sahlqvist's theorem [80] where two facts are proved for a large, syntactically defined class of modal formulae, now called Sahlqvist formulae: (i) All Sahlqvist formulae define first-order conditions on (standard relational) frames and these properties can be effectively computed (correspondence result), and (ii) all Sahlqvist formulae are canonical, i.e. valid in their canonical (relational) frames and hence axiomatise completely the classes of frames satisfying the corresponding first-order properties (completeness result). Sahlqvist and others have developed a variety of methods for reducing individual axioms to relational first-order correspondence properties [57, 80, 92]. While these methods are specific to modal correspondence theory (in the form described), there are also two generalised methods that can be used for computing correspondence properties from modal axioms and have been automated: SCAN [26, 29, 70] and DLS [24, 25, 91]. Both these methods provide algorithms for eliminating second-order quantifiers from second-order formulae. Interestingly both algorithms are based on a form of resolution. Here we focus just on the SCAN algorithm, but more details of DLS and other quantifier elimination algorithms can be found in [24, 66, 91].

Computing the first-order equivalents of modal formulae (if they exists) amounts to the elimination of universal or existential monadic second-order quantifiers. For example, if we are interested in establishing the relational frame properties corresponding to a modal axiom φ, then we either have to eliminate the universal monadic second-order quantifiers from $\forall\overline{Q_{p_i}}\forall x\, \pi_r(\varphi, x)$, or, equivalently, the existential monadic second-order quantifiers from

$$\Psi = \exists\overline{Q_{p_i}}\exists x\, \pi_r(\neg\varphi, x).$$

There can be no algorithm which is guaranteed to find a first-order equivalent formula if there exists one, i.e. no algorithm can be complete in general, but still SCAN (and DLS) provide a partial solution to the quantifier elimination problem. SCAN basically uses a special kind of constraint (or hierarchical) first-order resolution to generate logical consequences of the initial formula Ψ, eventually keeping from the resulting set of clauses only those in which none of the second-order variables occur. In general termination cannot be guaranteed. If the saturation process terminates then SCAN attempts to restore first-order quantifiers from the Skolem functions by reversing Skolemisation. If the reversal of Skolemisation succeeds then SCAN produces an equivalent first-order formula. This last step is not always possible, and in this case SCAN outputs an equivalent formula involving second-order Henkin quantifiers.

Experiences with SCAN are very positive. For example, it can compute the frame correspondence properties for very many well-known axioms such as *T*, *4*, *5*, and the others from Figure 3. In fact, recent work has shown:

Theorem 39 ([39]). *The* SCAN *algorithm is complete for the class of Sahlqvist formulae, in the sense that, when given a Sahlqvist formula, it successfully computes an equivalent first-order formula for it.*

Because of their generality the applicability of algorithms such as SCAN and DLS is not limited to correspondence theory with respect to the classical relational semantics. These algorithms, and in particular SCAN, which has a dedicated interface for modal logic, have proved useful in the investigations of non-standard translation methods, cf. e.g. [64, 69, 81], but the algorithms have also other applications.

10 Conclusion

Because we are interested in the big picture brought to view by the results and methods surveyed in this paper, and also due to lack of space and time, we have omitted many details and could not cover all important topics in the area. Thus we want to point the interested reader to the following survey papers: [27, 68, 69, 71] on various translation approaches for non-classical logics ([27] surveys also other methods including resolution methods), [66] on quantifier elimination algorithms, [46] on resolution and tableau approaches to modal logic, [56] on decidable first-order fragments relevant to description and modal logics, and [6, 7, 62] on the modern framework of resolution.

Let us conclude this survey with some remarks on the contributions of Harald Ganzinger to automated reasoning in modal logic. His *direct* contributions to modal logic include decision procedures for transitive modal logics based on the ordered chaining calculus [31] (cf. also [33]), and a generalisation of the SCAN algorithm of [29] as an instance of hierarchical theorem proving for first-order logic [9]. Also relevant is his work on solvable first-order fragments [8], and especially the work on the solvability of the guarded fragment and extensions with transitivity [30, 32]. For us the development and presentation of a uniform framework of resolution (and ultimately automated reasoning), together with Leo Bachmair [5–7], was, above anything, the most important contribution of Harald Ganzinger. Within this framework it is actually possible to obtain more general results than presented here, and the application and potential of the underlying principles and methods clearly go beyond just modal logics and related logics. The underlying principles and methods are quite general—universal even, as is more than amply demonstrated by past and present research in the area of automated reasoning.

References

1. H. Andréka, I. Németi, and J. van Benthem. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.
2. H. Andréka, J. van Benthem, and I. Németi. Back and forth between modal logic and classical logic. *Bulletin of the IGPL*, 3(5):685–720, 1995.
3. C. Areces, R. Gennari, J. Heguiabehere, and M. de Rijke. Tree-based heuristics in modal theorem proving. In *Proc. ECAI-2000*, pp. 199–203. IOS Press, 2000.
4. Y. Auffray and P. Enjalbert. Modal theorem proving: An equational viewpoint. *Journal of Logic and Computation*, 2(3):247–297, 1992.
5. L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994.
6. L. Bachmair and H. Ganzinger. Equational reasoning in saturation-based theorem proving. In W. Bibel and P. H. Schmitt, editors, *Automated Deduction—A Basis for Applications. Volume I*, pp. 353–397. Kluwer, 1998.
7. L. Bachmair and H. Ganzinger. Resolution theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, vol. I, pp. 19–99. Elsevier, 2001.

8. L. Bachmair, H. Ganzinger, and U. Waldmann. Superposition with simplification as a decision procedure for the monadic class with equality. In *Proc. KGC'93*, vol. 713 of *LNCS*, pp. 83–96. Springer, 1993.
9. L. Bachmair, H. Ganzinger, and U. Waldmann. Refutational theorem proving for hierarchic first-order theories. *Applicable Algebra in Engineering, Communication and Computing*, 5(3/4):193–212, 1994.
10. P. Baumgartner. FDPLL: A first-order Davis-Putnam-Logeman-Loveland procedure. In *Proc. CADE-17*, vol. 1831 of *LNAI*, pp. 200–219. Springer, 2000.
11. P. Baumgartner, J. D. Horton, and B. Spencer. Merge path improvements for minimal model hyper tableaux. In *Proc. TABLEAUX'99*, vol. 1617 of *LNAI*, pp. 51–65. Springer, 1999.
12. P. Baumgartner and R. A. Schmidt. Improved bottom-up model generation. Manuscript, submitted for publication, 2006.
13. P. Baumgartner and C. Tinelli. The model evolution calculus. In *Proc. CADE-19*, vol. 2741 of *LNAI*, pp. 350–364. Springer, 2003.
14. P. Blackburn, M. de Rijke, and V. Venema. *Modal Logic*, vol. 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001.
15. W. W. Bledsoe. Splitting and reduction heuristics in automatic theorem proving. *Artificial Intelligence*, 2:55–77, 1971.
16. C. Brink, K. Britz, and R. A. Schmidt. Peirce algebras. *Formal Aspects of Computing*, 6(3):339–358, 1994.
17. F. Bry and A. Yahya. Positive unit hyperresolution tableaux for minimal model generation. *Journal of Automated Reasoning*, 25(1):35–82, 2000.
18. M. A. Castilho, L. Fariñas del Cerro, O. Gasquet, and A. Herzig. Modal tableaux with propagation rules and structural rules. *Fundamenta Informaticae*, 3–4(32):281–297, 1997.
19. H. de Nivelle. Splitting through new proposition symbols. In *Proc. LPAR 2001*, vol. 2250 of *LNAI*, pp. 172–185. Springer, 2001.
20. H. de Nivelle, R. A. Schmidt, and U. Hustadt. Resolution-based methods for modal logics. *Logic Journal of the IGPL*, 8(3):265–292, 2000.
21. M. de Rijke. *Extending Modal Logic*. PhD thesis, University of Amsterdam, 1993.
22. S. Demri and H. de Nivelle. Deciding regular grammar logics with converse through first-order logic. *Journal of Logic, Language and Information*, 14(3):289–329, 2005.
23. S. Demri and D. Gabbay. On modal logics characterized by models with relative accessibility relations: Part II. *Studia Logica*, 66(3):349–384, 2000.
24. P. Doherty, W. Lukaszewicz, and A. Szalas. Computing circumscription revisited: A reduction algorithm. *Journal of Automated Reasoning*, 18(3):297–336, 1997.
25. P. Doherty, W. Lukaszewicz, A. Szalas, and J. Gustafsson. DLS, 1996. http://www.ida.liu.se/labs/kplab/projects/dls/.
26. T. Engel. Quantifier elimination in second-order predicate logic. Diplomarbeit, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, Germany, 1996.
27. L. Fariñas del Cerro and A. Herzig. Modal deduction with applications in epistemic and temporal logics. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming: Epistemic and Temporal Reasoning*, vol. 4, pp. 499–594. Clarendon Press, 1995.
28. C. Fermüller, A. Leitsch, U. Hustadt, and T. Tammet. Resolution decision procedures. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, vol. II, pp. 1791–1849. Elsevier, 2001.
29. D. M. Gabbay and H. J. Ohlbach. Quantifier elimination in second-order predicate logic. *South African Computer Journal*, 7:35–43, 1992.

30. H. Ganzinger and H. de Nivelle. A superposition decision procedure for the guarded fragment with equality. In *Proc. LICS-14*, pp. 295–303. IEEE, 1999.
31. H. Ganzinger, U. Hustadt, C. Meyer, and R. A. Schmidt. A resolution-based decision procedure for extensions of K4. In *Advances in Modal Logic, Volume 2*, vol. 119 of *Lecture Notes*, pp. 225–246. CSLI Publications, 2001.
32. H. Ganzinger, C. Meyer, and M. Veanes. The two-variable guarded fragment with transitive relations. In *Proc. LICS-14*, pp. 24–34. IEEE, 1999.
33. H. Ganzinger and V. Sofronie-Stokkermans. Chaining techniques for automated theorem proving in finitely-valued logics. In *Proc. ISMVL 2000*, pp. 337–344. IEEE, 2000.
34. G. Gargov and S. Passy. A note on Boolean modal logic. In *Mathematical Logic: Proceedings of the 1988 Heyting Summerschool*, pp. 299–309. Plenum Press, 1990.
35. G. Gargov, S. Passy, and T. Tinchev. Modal environment for Boolean speculations. In *Mathematical Logic and its Applications: Proceedings of the 1986 Gödel Conference*, pp. 253–263. Plenum Press, 1987.
36. L. Georgieva, U. Hustadt, and R. A. Schmidt. Computational space efficiency and minimal model generation for guarded formulae. In *Proc. LPAR 2001*, vol. 2250 of *LNAI*, pp. 85–99. Springer, 2001.
37. L. Georgieva, U. Hustadt, and R. A. Schmidt. A new clausal class decidable by hyperresolution. In *Proc. CADE-18*, vol. 2392 of *LNAI*, pp. 260–274. Springer, 2002.
38. L. Georgieva, U. Hustadt, and R. A. Schmidt. Hyperresolution for guarded formulae. *Journal of Symbolic Computation*, 36(1–2):163–192, 2003.
39. V. Goranko, U. Hustadt, R. A. Schmidt, and D. Vakarelov. SCAN is complete for all Sahlqvist formulae. In *Proc. RelMiCS 7*, vol. 3051 of *LNCS*, pp. 149–162. Springer, 2004.
40. R. Goré. Tableau methods for modal and temporal logics. In M. D'Agostino, D. Gabbay, R. Hähnle, and J. Posegga, editors, *Handbook of Tableau Methods*, pp. 297–396. Kluwer, 1999.
41. E. Grädel. Decision procedures for guarded logics. In *Proc. CADE-16*, vol. 1632 of *LNAI*, pp. 31–51. Springer, 1999.
42. E. Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 64:1719–1742, 1999.
43. R. Hasegawa, H. Fujita, and M. Koshimura. Efficient minimal model generation using branching lemmas. In *Proc. CADE-17*, vol. 1421 of *LNAI*, pp. 184–199. Springer, 2000.
44. A. Herzig. *Raisonnement automatique en logique modale et algorithmes d'unification.* PhD thesis, University Paul-Sabatier, Toulouse, 1989.
45. A. Herzig. A new decidable fragment of first order logic, 1990. In Abstracts of 3rd Logical Biennial, Summer School & Conf. in honour of S. C. Kleene, Bulgaria.
46. I. Horrocks, U. Hustadt, U. Sattler, and R. A. Schmidt. Computational modal logic. In P. Blackburn, J. van Benthem, and F. Wolter, editors, *Handbook of Modal Logic*. Elsevier, 2006. To appear.
47. I. L. Humberstone. Inaccessible worlds. *Notre Dame Journal of Formal Logic*, 24(3):346–352, 1983.
48. I. L. Humberstone. The modal logic of 'all and only'. *Notre Dame Journal of Formal Logic*, 28(2):177–188, 1987.
49. U. Hustadt. *Resolution-Based Decision Procedures for Subclasses of First-Order Logic.* PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 1999.
50. U. Hustadt and R. A. Schmidt. An empirical analysis of modal theorem provers. *Journal of Applied Non-Classical Logics*, 9(4):479–522, 1999.

51. U. Hustadt and R. A. Schmidt. Maslov's class K revisited. In *Proc. CADE-16*, vol. 1632 of *LNAI*, pp. 172–186. Springer, 1999.

52. U. Hustadt and R. A. Schmidt. On the relation of resolution and tableaux proof systems for description logics. In *Proc. IJCAI'99*, vol. 1, pp. 110–115. Morgan Kaufmann, 1999.

53. U. Hustadt and R. A. Schmidt. Issues of decidability for description logics in the framework of resolution. In *Automated Deduction in Classical and Non-Classical Logics*, vol. 1761 of *LNAI*, pp. 191–205. Springer, 2000.

54. U. Hustadt and R. A. Schmidt. MSPASS: Modal reasoning by translation and first-order resolution. In *Proc. TABLEAUX 2000*, vol. 1847 of *LNAI*, pp. 67–71. Springer, 2000.

55. U. Hustadt and R. A. Schmidt. Using resolution for testing modal satisfiability and building models. *Journal of Automated Reasoning*, 28(2):205–232, 2002.

56. U. Hustadt, R. A. Schmidt, and L. Georgieva. A survey of decidable first-order fragments and description logics. *Journal of Relational Methods in Computer Science*, 1:251–276, 2004.

57. M. Kracht. *Tools and Techniques in Modal Logic*, vol. 142 of *Studies in Logic*. Elsevier, 1999.

58. S. Ju. Maslov. The inverse method for establishing deducibility for logical calculi. In V. P. Orevkov, editor, *The Calculi of Symbolic Logic I: Proc. of the Steklov Institute of Mathematics edited by I. G. Petrovskiĭ and S. M. Nikol'skiĭ, Nr. 98 (1968)*, pp. 25–96. Amer. Math. Soc., Providence, Rhode Island, 1971.

59. F. Massacci. Single step tableaux for modal logics: Computational properties, complexity and methodology. *Journal of Automated Reasoning*, 24(3):319–364, 2000.

60. K. Nellas. Reasoning about sets and relations: A tableaux-based automated theorem prover for Peirce logic. Master's thesis, University of Manchester, UK, 2001.

61. I. Niemelä. A tableau calculus for minimal model reasoning. In *Proc. TABLEAUX'96*, vol. 1071 of *LNAI*, pp. 278–294. Springer, 1996.

62. Robert Nieuwenhuis and Albert Rubio. Paramodulation-based theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, vol. I, pp. 371–443. Elsevier, 2001.

63. A. Nonnengart. First-order modal logic theorem proving and functional simulation. In *Proc. IJCAI'93*, pp. 80–85. Morgan Kaufmann, 1993.

64. A. Nonnengart. *A Resolution-Based Calculus For Temporal Logics*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 1995.

65. A. Nonnengart. Resolution-based calculi for modal and temporal logics. In *Proc. CADE-13*, vol. 1104 of *LNAI*, pp. 598–612. Springer, 1996.

66. A. Nonnengart, H. J. Ohlbach, and A. Szalas. Quantifier elimination for second-order predicate logic. To appear in *Logic, Language and Reasoning: Essays in honour of Dov Gabbay*, Part I, Kluwer.

67. H. J. Ohlbach. Semantics based translation methods for modal logics. *Journal of Logic and Computation*, 1(5):691–746, 1991.

68. H. J. Ohlbach. Translation methods for non-classical logics: An overview. *Bulletin of the IGPL*, 1(1):69–89, 1993.

69. H. J. Ohlbach. Combining Hilbert style and semantic reasoning in a resolution framework. In *Proc. CADE-15*, vol. 1421 of *LNAI*, pp. 205–219. Springer, 1998.

70. H.-J. Ohlbach and T. Engel. SCAN, 1994. http://www.mpi-inf.mpg.de/departments/d2/software/SCAN/.

71. H. J. Ohlbach, A. Nonnengart, M. de Rijke, and D. Gabbay. Encoding two-valued nonclassical logics in classical logic. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, vol. II, pp. 1403–1486. Elsevier, 2001.
72. H. J. Ohlbach and R. A. Schmidt. Functional translation and second-order frame properties of modal logics. *Journal of Logic and Computation*, 7(5):581–603, 1997.
73. W. C. Purdy. Decidability of fluted logic with identity. *Notre Dame Journal of Formal Logic*, 37(1):84–104, 1996.
74. W. C. Purdy. Quine's 'limits of decision'. *Journal of Symbolic Logic*, 64(4):1439–1466, 1999.
75. W. V. Quine. Variables explained away. In *Proc. American Philosophy Society*, vol. 104, pp. 343–347, 1960.
76. W. V. Quine. Algebraic logic and predicate functors. In R. Rudner and I. Scheffler, editors, *Logic and Art: Esssays in Honor of Nelson Goodman*. Bobbs-Merrill, 1971.
77. A. Riazanov and A. Voronkov. Vampire. In *Proc. CADE-16*, vol. 1632 of *LNAI*, pp. 292–296. Springer, 1999.
78. A. Riazanov and A. Voronkov. Splitting without backtracking. In *Proc. IJCAI 2001*, pp. 611–617. Morgan Kaufmann, 2001.
79. J. A. Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12(1):23–41, 1965.
80. H. Sahlqvist. Completeness and correspondence in the first and second order semantics for modal logics. In *Proc. 3rd Scandinavian Logic Symposium, 1973*, pp. 110–143. North-Holland, 1975.
81. R. A. Schmidt. *Optimised Modal Translation and Resolution*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 1997.
82. R. A. Schmidt. Decidability by resolution for propositional modal logics. *Journal of Automated Reasoning*, 22(4):379–396, 1999.
83. R. A. Schmidt. MSPASS, 1999. http://www.cs.man.ac.uk/˜schmidt/mspass/.
84. R. A. Schmidt and U. Hustadt. A resolution decision procedure for fluted logic. In *Proc. CADE-17*, vol. 1831 of *LNAI*, pp. 433–448. Springer, 2000.
85. R. A. Schmidt and U. Hustadt. Mechanised reasoning and model generation for extended modal logics. In *Theory and Applications of Relational Structures as Knowledge Instruments*, vol. 2929 of *LNCS*, pp. 38–67. Springer, 2003.
86. R. A. Schmidt and U. Hustadt. A principle for incorporating axioms into the first-order translation of modal formulae. In *Proc. CADE-19*, vol. 2741 of *LNAI*, pp. 412–426. Springer, 2003.
87. R. A. Schmidt and U. Hustadt. The axiomatic translation principle for modal logic. *ACM Transactions on Computational Logic*, 2005. To appear.
88. R. A. Schmidt, E. Orlowska, and U. Hustadt. Two proof systems for Peirce algebras. In *Proc. RelMiCS 7*, vol. 3051 of *LNCS*, pp. 238–251. Springer, 2004.
89. S. Schulz. E: A brainiac theorem prover. *Journal of AI Communications*, 15(2–3):111–126, 2002.
90. G. Stenz. Dctp 1.2: System abstract. In *Proc. TABLEAUX 2002*, vol. 2381 of *LNAI*, pp. 335–340. Springer, 2002.
91. A. Szałas. On the correspondence between modal and classical logic: An automated approach. *Journal of Logic and Computation*, 3(6):605–620, 1993.
92. J. van Benthem. Correspondence theory. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic*, vol. II, pp. 167–247. Reidel, Dordrecht, 1984.
93. C. Weidenbach, U. Brahm, T. Hillenbrand, E. Keen, C. Theobald, and D. Topic. SPASS version 2.0. In *Proc. CADE-18*, vol. 2392 of *LNAI*, pp. 275–279. Springer, 2002.

94. N. K. Zamov. Modal resolutions. *Soviet Mathematics*, 33(9):22–29, 1989. Translated from *Izv. Vyssh. Uchebn. Zaved. Mat.* **9** (328) (1989) 22–29.