A Resolution-Based Decision Procedure for Extensions of K4

H. Ganzinger[1], U. Hustadt[2], C. Meyer[1,⋆], and R. A. Schmidt[2]

[1] Max-Planck-Institut für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany,
hg@mpi-sb.mpg.de, meyer@mpi-sb.mpg.de

[2] Department of Computing and Mathematics, Manchester Metropolitan University,
Chester Street, Manchester M1 5GD, UK,
R.A.Schmidt@doc.mmu.ac.uk, U.Hustadt@doc.mmu.ac.uk

Abstract. This paper presents a resolution decision procedure for transitive propositional modal logics. The procedure combines the relational translation method with an ordered chaining calculus designed to avoid unnecessary inferences with transitive relations. We show the logics K4, KD4 and S4 can be transformed into a bounded class of well-structured clauses closed under ordered resolution and negative chaining.
Area: Computational aspects of modal logic

1 Introduction

The iterated modality in the schema $4 = \Box p \rightarrow \Box\Box p$ is cause for some difficulties. Because the number of modal operators does not diminish during deduction in Hilbert calculi, tableaux-like calculi or modal resolution calculi, as they would in systems, like K, KD or KT, in order to avoid unlimited derivations, some form of cycle detection mechanism is essential.

Semantics-based translation approaches have similar problems. Translation approaches are based on the idea that modal inference can be done by translating modal formulae into first-order logic and conventional first-order theorem proving. Here the difficulty is caused by

$$\text{transitivity:} \quad \forall x, y, z\ (x\,R\,y \wedge y\,R\,z) \rightarrow x\,R\,z$$

which leads, in general, to unlimited growth of the size of formulae. A makeshift solution for the optimised functional translation method uses pre-computed term depth bounds, whereby termination can be guaranteed [18,16]. However, in practice this solution is very poor [11]. For non-transitive modal logics, good performance results have been obtained with the resolution theorem prover SPASS [8,11,9,18].

⋆ Corresponding author

Transitive relations play also an important role in automated deduction for first-order logic. A general resolution calculus designed for binary relations satisfying the general scheme $R_i \circ R_j \subseteq R_k$ (including equality) is by Bachmair and Ganzinger [3], and combines ideas from rewrite systems and resolution in a calculus of ordered chaining.

In this paper we show how this calculus may be used to obtain resolution decision procedures for the relational translation of a range of propositional modal logics. For the purpose of clarity we will focus on K4, KD4, and S4. The method may be applied also to multi-modal logics with modal operators satisfying (a subset of) D, T, and 4 as well as combinations thereof. The important ingredients of our method are structural transformation and ordered chaining with selection. Structural transformation allows us to embed the logics and formulae under consideration into a well-behaved class of clauses.

This paper is both of theoretical and practical interest, for modal logic as well as for automated deduction. Of interest to modal logic is that our method provides a new inference method for extensions of K4. Mechanisms like cycle detection as used in tableaux calculi are not required. There is also no need to go through the search space determined by a pre-computed proof depth bound. Our solution requires no specialised techniques, only standard theorem proving techniques are used. In particular, the chaining calculus is parameterised by an ordering and a selection function. The whole effort has been to find a suitable ordering and selection function so as to ensure termination for extensions of K4. Soundness and completeness of this application follows from soundness and completeness of the general chaining calculus. Of interest to automated deduction is that the ordered chaining calculus, which has been developed to overcome problems of traditional approaches in automated theorem proving with transitivity axioms, also provides a basis for the development of decision procedures for subclasses of first-order logic with transitivity.

The structure of the paper is as follows. Section 2 gives some preliminary definitions and notation. Section 3 defines the modal logics under consideration, the relational translation to first-order logic, and a structural transformation of first-order formulae. The examples in Section 4 illustrate the causes of non-termination by unrefined resolution. Our decision procedure is based on the ordered chaining calculus defined in Section 5. Section 6 describes a finitely bounded class of clauses which includes the input clauses stemming from the translation of modal formulae. Section 7 proves that this class is closed under inferences in the chaining calculus with eager condensation and proves termination. The examples of Section 8 provides a sample refutation and illustrate how ordered chaining avoids indefinite computations. The examples also show that our method is different from tableaux methods. The final section discusses further work.

2 Preliminary definitions and notation

In addition to propositional modal logics we consider first-order languages with function symbols, predicate symbols, and variables. A *term* is an expression $f(t_1, \ldots, t_n)$, where f is a function symbol of arity n and $t_1, \ldots, t_n$ are terms, or a variable x. An *atom* is an expression $P(t_1, \ldots, t_n)$ where P is a predicate symbol of arity n and $t_1, \ldots, t_n$ are terms. A *literal* is an atom (a *positive literal*) or its negation $\neg A$ (a *negative literal*). Atoms with binary predicate symbol R will be written in infix notation, for example $s\,R\,t$. Clauses are finite multisets of literals and will be written as disjunctions. Two atoms (literals or clauses) are *variants* of each other, if they are equal modulo renaming of variables.

A *position* is a word over the natural numbers. The set $\mathrm{Pos}(\phi)$ of positions of a given formula ϕ is defined by: (i) the empty word $\epsilon \in \mathrm{Pos}(\phi)$, (ii) $i.\lambda \in \mathrm{Pos}(\phi)$ if $\phi = \phi_1 \star \cdots \star \phi_n$ and $\lambda \in \mathrm{Pos}(\phi)$, where $\star$ is a first-order connective (quantifiers $\forall x$ and $\exists x$ are regarded to be unary connectives). If λ is a position in ϕ, then $\phi|_\lambda$ denotes the subformula of ϕ at position λ, that is, $\phi|_\epsilon = \phi$ and $\phi|_{i.\lambda} = \phi_i|_\lambda$. The result of replacing ϕ at position λ by ψ is denoted by $\phi[\lambda \leftarrow \psi]$. We write $\phi[\psi]_\lambda$ for ϕ when $\phi_\lambda = \psi$.

An occurrence of a subformula has *positive polarity* if it occurs inside the scope of an even number of (explicit or implicit) negations, and an occurrence has *negative polarity* if it occurs inside the scope of an odd number of negations. For example, both occurrences of the subformula $\neg C \wedge D$ in $\Diamond(\neg C \wedge D) \wedge \Box(\neg C \wedge D)$ have positive polarity, and both occurrences of C have negative polarity.

The following notation will be adopted. First-order variables are denoted by $x, y, z, x', y', z', \ldots$, terms by $s, t, u, v, s', t', u', v', \ldots$, atoms by $A, B, A', B', \ldots$, and clauses by $C, D, C', D', \ldots$. By $\mathrm{V}(C)$ we denote the set of free variables occurring in C.

3 The relational translation of modal formulae

The language of the propositional modal logic $\mathsf{K}\Sigma$ is that of propositional logic plus additional modal operators $\Box$ and $\Diamond$. By definition, a *formula* of $\mathsf{K}\Sigma$ is a Boolean combination of propositional and modal atoms. A *modal atom* is an expression of the form $\Box\psi$ or $\Diamond\psi$ where ψ is a formula of $\mathsf{K}\Sigma$. A *literal* is a propositional atom or its negation. In the following we assume that modal formulae are in negation normal form, containing no occurrences of the Boolean connectives $\rightarrow$ (implication) and $\leftrightarrow$ (equivalence). In general, Σ is a (possibly empty) set of additional frame properties which need not be modally definable. We assume Σ includes transitivity and possibly also frame properties from Table 1.

The aim is to show the satisfiability of a modal formula ϕ in a logic $\mathsf{K}\Sigma$. We will do so by refuting the translation of ϕ, and the translation we will use is the standard relational translation.

By definition, the relational translation operator Π_r^Σ maps ϕ to

$$\mathrm{Ax}_\Sigma \wedge \exists w\ \pi_r(\phi, w),$$

where Ax_Σ is the conjunction of first-order formulae corresponding to frame properties in Σ. The morphism π_r is defined by

$$\begin{aligned}
\pi_r(p, x) &= P(x) \\
\pi_r(\neg p, x) &= \neg P(x) \\
\pi_r(\phi_1 \wedge \cdots \wedge \phi_n, x) &= \pi_r(\phi_1, x) \wedge \cdots \wedge \pi_r(\phi_n, x) \\
\pi_r(\phi_1 \vee \cdots \vee \phi_n, x) &= \pi_r(\phi_1, x) \vee \cdots \vee \pi_r(\phi_n, x) \\
\pi_r(\Box\phi, x) &= \forall y\ (x\,R\,y \rightarrow \pi_r(\phi, y)) \\
\pi_r(\Diamond\phi, x) &= \exists y\ (x\,R\,y \wedge \pi_r(\phi, y)).
\end{aligned}$$

p is a propositional variable and P is a unary predicate uniquely associated with p. The symbol R is a special binary predicate denoting the accessibility relation in the underlying Kripke semantics. As $\pi_r(\phi, x)$ is in negation normal form, all non-atomic subformulae of $\pi_r(\phi, x)$ have positive polarity.

In order to obtain a simple clausal form we make use of a particular form of *structural transformation*. The idea of structural transformation is to introduce for particular subformula occurrences ψ of a formula ϕ a new 'name' Q_ψ. Such transformations were used by Tseitin [19] in studying the relative complexity of proof systems of propositional logic. They form

Seriality	D	$\Box p \rightarrow \Diamond p$	$\forall x \exists y\ x\,R\,y$
Reflexivity	T	$\Box p \rightarrow p$	$\forall x\ x\,R\,x$
Irreflexivity			$\forall x\ \neg(x\,R\,x)$
Transitivity	4	$\Box p \rightarrow \Box\Box p$	$\forall x, y, z\ (x\,R\,y \wedge y\,R\,z) \rightarrow x\,R\,z$

Table 1. Frame properties

a standard technique not only in the connection with resolution decision procedures [6,10,17], but allow also linear transformation of first-order formulae into clausal form [2,5,15].

Let P be a subset of $\mathrm{Pos}(\phi)$ for a first-order formula ϕ in negation normal form. We associate with each position λ of P a new predicate symbol Q_λ and a new literal $Q_\lambda(x_1,\ldots,x_n)$, where $x_1, \ldots, x_n$ are the free variables of $\phi|_\lambda$. The definition $\mathrm{Def}_\lambda(\phi)$ of Q_λ is the formula

$$\forall x_1,\ldots,x_n\ (Q_\lambda(x_1,\ldots,x_n) \rightarrow \phi|_\lambda).$$

Define $\mathrm{Def}_P(\phi)$ inductively by:

$$\begin{aligned}\mathrm{Def}_\emptyset(\phi) &= \phi\\ \mathrm{Def}_{P\cup\{\lambda\}}(\phi) &= \mathrm{Def}_P(\mathrm{Def}_\lambda(\phi) \wedge \phi[\lambda \leftarrow Q_\lambda(x_1,\ldots,x_n)]),\end{aligned}$$

where λ is a maximal element of $C \cup \{\lambda\}$ with respect to the prefix ordering on positions.

Since the mapping π_r preserves the structure of a formula, we can associate with each position of a modal formula ϕ a unique position in $\pi_r(\phi,w)$. Let $\mathrm{Pos}^+_m(\phi)$ be the set of positions of non-literal subformulae of a modal formula ϕ. Let $\mathrm{Pos}^+_r(\phi)$ be the set of positions in $\pi_r(\phi,w)$ associated with $\mathrm{Pos}^+_m(\phi)$. By Ξ we denote the transformation taking $\Pi^\Sigma_r(\phi)$ to

$$\Pi^\Sigma_r(\phi)|_1 \wedge \exists w\ \mathrm{Def}_{\mathrm{Pos}^+_r(\phi)}(\Pi^\Sigma_r(\phi)|_{2.1})$$

where $\Pi^\Sigma_r(\phi)|_{2.1} = \pi_r(\phi,w)$ and $\Pi^\Sigma_r(\phi)|_1 = \mathrm{Ax}_\Sigma$.

Theorem 1. *Let ϕ be modal formula in negation normal form. Then:*

1. *ϕ is $\mathsf{K}\Sigma$-satisfiable if and only if $\Pi^\Sigma_r(\phi)$ is a satisfiable first-order formula.*
2. *$\Pi^\Sigma_r(\phi)$ is a satisfiable first-order formula if and only if $\Xi\Pi^\Sigma_r(\phi)$ is a satisfiable first-order formula.*

Let $\mathrm{Cls}(\phi)$ denote the clausification of a first-order formula ϕ, which is computed by removing all occurrences of the logical connectives $\rightarrow$, Skolemisation, and turning the Skolemised formula into clausal form. In this paper we assume that outer or inner Skolemisation is used, but for Strong Skolemisation [14] the decision procedure and decidability result are the same.

Using Theorem 1, we can show the satisfiability and unsatisfiability of a modal formula ϕ by showing the satisfiability of the first-order formula $\Xi\Pi^\Sigma_r(\phi)$ which in turn can be realised by showing the satisfiability of $\mathrm{Cls}\Xi\Pi^\Sigma_r(\phi)$.

4 Examples

The following example shows that even for the translation of the basic modal logic K unrefined resolution is only a semi-decision procedure.

Consider $\Pi_r^{\mathsf{T4}}(\Box\Diamond p) =$

$$\begin{aligned}&\forall x\ x\,R\,x\\ &\wedge\ \forall x,y,z\ (x\,R\,y \wedge y\,R\,z) \rightarrow x\,R\,z)\\ &\wedge\ \exists x\ (\forall y\ (x\,R\,y \rightarrow \exists z\ (y\,R\,z \wedge P(z)))).\end{aligned}$$

The result of applying Ξ is

$$\begin{aligned}&\forall x\ x\,R\,x\\ &\wedge\ \forall x,y,z\ (x\,R\,y \wedge y\,R\,z) \rightarrow x\,R\,z\\ &\wedge\ \exists x\ Q_2(x)\\ &\wedge\ \forall x\ (Q_2(x) \rightarrow (\forall y\ x\,R\,y \rightarrow Q_1(y)))\\ &\wedge\ \forall x\ (Q_1(x) \rightarrow (\exists y\ x\,R\,y \wedge P(y))).\end{aligned}$$

From the last two conjuncts we obtain the following three clauses.

$$\neg Q_2(x) \vee \neg(x\,R\,y) \vee Q_1(y)$$

$$\neg Q_1(x) \vee x\,R\,f_1(x)$$

$$\neg Q_1(x) \vee P(f_1(x))$$

By resolving on the literals marked in gray, we get the following non-terminating derivation.

$$\neg Q_1(x) \vee \neg Q_2(x) \vee Q_1(f_1(x))$$

$$\neg Q_1(x) \vee \neg Q_2(x) \vee f_1(x)\,R\,f_1(f_1(x))$$

$$\neg Q_1(x) \vee \neg Q_2(x) \vee \neg Q_2(f_1(x)) \vee Q_1(f_1(f_1(x)))$$

The problem is the unbounded growth of the depth of terms and the unbounded growth of the number of literals in the clauses in the derivation.

The standard approach for defining decision procedures based on resolution for decidable classes of first-order formulae makes use of *refinements* of resolution. These refinements, whilst preserving the soundness and completeness of resolution, are used to constrain the possible inferences by resolution in such a way that both (i) the growth of the depth of terms in resolvents, and (ii) the growth of the number of literals in resolvents is bounded in any derivation. Then any derivation eventually terminates. The most commonly used refinements are *ordering refinements* [7,12,6,10]. Notably, ordering refinements have been in the focus of research on automated deduction by resolution independent of the consideration of decidability issues [4].

Transitivity did not come into the above derivation. In the presence of transitivity the formula $\Box p$ results in an infinite derivation.

$$\neg(x\,R\,y) \vee \neg(y\,R\,z) \vee \boxed{x\,R\,z}$$
$$\neg Q_1(x) \vee \boxed{\neg(x\,R\,y)} \vee P(y)$$
$$\neg Q_1(x) \vee \boxed{\neg(x\,R\,y)} \vee \neg(y\,R\,z) \vee P(z)$$
$$\neg Q_1(x) \vee \boxed{\neg(x\,R\,y')} \vee \neg(y'\,R\,y) \vee \neg(y\,R\,z) \vee P(z)$$
$$\vdots$$

The first clause is the transitivity clause, the second clause represents $\Box p$, and the remaining clauses are derived clauses.

5 Ordered chaining

We recall the definition of the *ordered chaining calculus* C from [3]. As usual we implicitly assume that the premises of an inference have no common variables. If necessary, the variables in one premise are renamed. Thus, it is also possible to use different variants of a clause as premises in one inference.

The calculus is parameterised by a certain class of well-founded orderings $\succ$ on ground terms and literals and by selection functions S. On ground terms $\succ$ has to be a total reduction ordering. On literals the ordering must be "admissible" in the sense defined in [3]. A particular such ordering will be given in the Section 7 below. A *selection* function assigns to each ground clause a possibly empty set of (occurrences of) negative literals. If C is a ground clause, then the literal occurrences in $S(C)$ are *selected*. The inference rules are restricted by constraints involving the specific ordering and selection function that one is using. For this purpose, the ordering is lifted to non-ground expressions E, E' as follows: $E \succ E'$ if and only if $E\sigma \succ E'\sigma$, for all ground substitutions σ.

Constraints about selection in an inference involving a (non-ground) premise C have the following meaning. A literal L is *selected in* C whenever there exists a ground instance (by a substitution σ) of the inference such that the ordering constraints are satisfied and such that the corresponding occurrence $L\sigma$ of L is selected in $C\sigma$. Conversely, a literal L is *not selected* in C whenever there is no such ground instance of the inference such that $L\sigma$ is selected in C.

These are the inference rules of the calculus:

Ordered resolution:

$$\frac{C \vee A \qquad D \vee \neg B}{C\sigma \vee D\sigma}$$

where σ is the most general unifier of A and B, $A\sigma$ is strictly maximal with respect to $C\sigma$, no literal is selected in C, and either $\neg B$ is selected or else $\neg B\sigma$ is maximal with respect to $D\sigma$. We call $C \vee A$ the *positive premise* and $D \vee \neg B$ the *negative premise*.

Ordered factoring:

$$\frac{C \vee A \vee B}{C \vee A}$$

where σ is the most general unifier of A and B, $A\sigma$ is maximal with respect to $C\sigma$, and no literal is selected in C. Factoring of negative clauses is not necessary for completeness.

Ordered chaining:

$$\frac{C \vee u\,R\,s \qquad D \vee t\,R\,v}{C\sigma \vee D\sigma \vee u\sigma\,R\,v\sigma}$$

where σ is the most general unifier of s and t, $u\sigma\,R\,s\sigma$ is strictly maximal with respect to $C\sigma$, $t\sigma\,R\,v\sigma$ is strictly maximal with respect to $D\sigma$, $u\sigma \not\succeq s\sigma$, $v\sigma \not\succeq t\sigma$, and no literal is selected in C and D.

Negative chaining:

$$\frac{C \vee \neg(u\,R\,s) \qquad D \vee t\,R\,v}{C\sigma \vee D\sigma \vee \neg(v\sigma\,R\,s\sigma)}$$

where σ is the most general unifier of u and t, $t\sigma\,R\,v\sigma$ is strictly maximal with respect to $D\sigma$, no literal is selected in D, either $\neg(u\,R\,s)$ is selected or else $\neg(u\sigma\,R\,s\sigma)$ is maximal with respect to $C\sigma$, $v\sigma \not\succeq t\sigma$, $s\sigma \not\succeq u\sigma$, and $s\sigma \neq v\sigma$ and

$$\frac{C \vee \neg(u\,R\,s) \qquad D \vee t\,R\,v}{C\sigma \vee D\sigma \vee \neg(u\sigma\,R\,t\sigma)}$$

where σ is the most general unifier of s and v, $t\sigma\,R\,v\sigma$ is strictly maximal with respect to $D\sigma$, no literal is selected in D, either $\neg(u\,R\,s)$ is selected or else $\neg(u\sigma\,R\,s\sigma)$ is maximal with respect to $C\sigma$, $t\sigma \not\succeq v\sigma$, $u\sigma \not\succ s\sigma$, and $u\sigma \neq t\sigma$.

Ordered chaining and negative chaining are macro inferences with the transitivity clause for R. Given ground clauses $C \vee u\,R\,s$ and $D \vee s\,R\,v$ we can derive $C \vee D \vee u\,R\,v$ by resolving with the first and second literal of $\neg(x\,R\,y) \vee \neg(y\,R\,z) \vee x\,R\,z$. Given ground clauses $C \vee \neg(u\,R\,s)$ and

$D \vee uRv$ we may derive $C \vee D \vee \neg(v \, R \, s)$ by resolving with the first and third literal of $\neg(x \, R \, y) \vee \neg(y \, R \, z) \vee x \, R \, z$ with unifier $\{x/u, y/v, z/s\}$. In a similar way the second form of negative chaining is justified.

With the ordering restrictions on the rules the explicit generation of the full transitive closure of R can usually be avoided. The intuition behind the way the ordering restrictions work arises from standard techniques in term rewriting. Suppose there is a transitive relation R given by $s \, R \, t$ and $t \, R \, u$ where s, t, u are ground terms. The ordered chaining inference rule is restricted such that $s \, R \, u$ is derived only if $t \succ s$ and $t \succ u$. This situation is called a *peak*. If $s \, R \, u$ has been computed the corresponding peak is said to *commute*. A system in which any peak commutes has properties similar to a convergent rewrite system in the equational case. In particular, the search for proofs of R facts can be restricted to so-called *rewrite proofs*. The ordering restrictions for negative chaining are designed to exclude the enumeration of proofs which are not rewrite proofs.

Note that the accessibility relation R in our fragment is *monotone* in that the representation of R by first-order terms always grows in size from one world to the next world. The ordering restriction of the chaining inferences exploit this structure and thus avoid many useless inferences.

Depending on certain technical details that we cannot discuss here, one additional inference is needed for certain occurrences of disjunctions of positive R literals:

Transitivity resolution:

$$\frac{C \vee s \, R \, u \vee t \, R \, v}{C\sigma \vee \neg(u\sigma \, R \, v\sigma) \vee s\sigma \, R \, v\sigma}$$

where σ is the most general unifier of s and t, $s\sigma \, R \, u\sigma$ is strictly maximal with respect to $C\sigma$, $t\sigma \, R \, v\sigma$ is maximal with respect to $C\sigma$, $u\sigma \not\succeq s\sigma$, $v\sigma \not\succeq t\sigma$, and no literal is selected in C.

The calculus is refutationally complete and compatible with a certain notion of *redundancy* for clauses and inferences by which additional don't-care non-deterministic simplification and deletion techniques can be justified. We do not want to repeat the formal definitions from [3]. Assuming this definition of redundancy we say that a set of clauses is *saturated up to redundancy* (with respect to ordered chaining) if the conclusion of every inference from non-redundant premises in N is either contained in N, or else is redundant in N.

Theorem 2 ([3]). *Let N be saturated up to redundancy. Then either N is satisfiable or else N contains the empty clause.*

The replacement of a clause by a condensed variant is covered by the notion of redundancy. We say that C is the *condensation of* $C \vee D$ if σ is a substitution such that $(C \vee D)\sigma$ and C contain the same *set* of literals and C is a minimal (with respect to size) proper subclause of $C \vee D$.

Proposition 3. *If C is a proper subclause of $C \vee D$ then $C \vee D$ is redundant in $N \cup \{C\}$, for any set N of clauses.*

The proof of this fact can be easily checked from the definition of redundancy in [3]. The significance of this proposition is that whenever an inference is to be computed, one may instead add the condensation of its conclusion to the current set of clauses. This is sound, and subsequently the conclusion of the inference becomes redundant. The completeness theorem only requires to saturate a set of clauses up to redundancy which, in turn, only requires that those conclusions of inferences be present which are not redundant.

The remainder of the paper is devoted to the definition of a class of condensed clauses that (i) is finite whenever the signature is finite; (ii) the demonstration that input clauses from the structural translation are within this class; and (iii) the proof of closure of this class under the inferences of the ordered chaining calculus with eager condensation, provided an adequate ordering and selection function is employed.

Under these circumstances, ordered chaining becomes an effective decision procedure for the modal logics we consider. Note that the constraints which restrict the inferences may or may not be decidable. In particular, the lifting of the ordering to non-ground expression involves universal quantification over all ground substitutions. Therefore all we can expect for any implementation of the calculus is the availability of a sound, decidable *approximation* of the constraints such that whenever a constraint is classified as unsatisfiable it is, in fact, unsatisfiable, while the converse need not be the case. However, how crude the approximation may be, having proved (iii) once and for all, we simply may ignore any inference which passes the approximative constraint check but produces a clause outside of the class of condensed clauses.

6 A class of clauses

Structural transformation by Ξ ensures that clauses in $\mathrm{Cls}\Xi\Pi_r(\phi)$ have a characteristic structure. The definition of an occurrence of a $\Box$ formula is represented by a clause of the form

$$\neg Q_i(x) \vee \neg(x\,R\,y) \vee (\neg)P(y). \tag{1}$$

A definition introduced for an occurrence of a $\Diamond$ formula generates clauses of the form

(2) $\neg Q_i(x) \vee x\, R\, f_i(x)$

(3) $\neg Q_i(x) \vee (\neg)P(f_i(x)),$

where f_i is a unary Skolem function which is uniquely associated with the renaming predicate Q_i. A definition introduced for a disjunction $(\neg)P_1 \vee \ldots \vee (\neg)P_n$ generates a clause of the form

(4) $\neg Q_i(x) \vee (\neg)P_1(x) \vee \ldots \vee (\neg)P_n(x).$

For a conjunction $(\neg)P_1 \wedge \ldots \wedge (\neg)P_n$ we obtain a set of clauses

$$\begin{array}{l} \neg Q_i(x) \vee (\neg)P_1(x) \\ \vdots \\ \neg Q_i(x) \vee (\neg)P_n(x), \end{array}$$

which are special cases of (4). Note that clauses of the form (1) to (4) contain at least one negative literal. In addition one positive unit clause is produced

(5) $Q_k(\iota)$

where ι is a Skolem constant. Finally, we have the clauses resulting from the transformation of Ax_Σ to clausal normal form, except that we delete the transitivity clause. For example, we will consider the reflexivity clause

(6) $x\, R\, x$

and the seriality clause

(7) $x\, R\, f(x).$

We introduce some more notation to abbreviate certain more general forms of clauses. Subsequently we assume that:

$$\begin{array}{rll} \neg(\bar{x}_n\, R\, t) & \text{expands to} & \bigvee_{1\leq i\leq n} \neg(x_i\, R\, t), \\ \neg(t\, R\, \bar{x}_n) & \text{expands to} & \bigvee_{1\leq i\leq n} \neg(t\, R\, x_i), \\ \mathcal{P}(\bar{x}_n) & \text{expands to} & \bigvee_{1\leq i\leq n} \mathcal{P}(x_i), \quad \text{and} \\ \mathcal{P}(t) & \text{expands to} & (\neg)P_1(t) \vee \ldots \vee (\neg)P_m(t), \end{array}$$

where t is a term and $\bar{x}_n$ denotes a vector of variables. If the number of variables is not important we write $\bar{x}$ instead of $\bar{x}_n$. Any of the disjunctions may be empty. The P_i in $\mathcal{P}(t)$ are pairwise distinct monadic predicates applied to the same term t. Different occurrences of $\mathcal{P}$ within a clause may involve different sets of predicates. For an example, let $\bar{x}_2$ be the vector of two variables, x_1 and x_2, and assume that there are two monadic predicates P and Q. Then $\mathcal{P}(\bar{x}_2) \vee \mathcal{P}(a)$ may expand to a clause $P(x_1) \vee \neg P(x_2) \vee Q(x_1) \vee Q(a)$, but not to $P(x_1) \vee P(x_1) \vee Q(a)$.

In generalising the types (1), (3), and (4), we arrive at the class of clauses C

(8) $$\mathcal{P}(\bar{x}) \vee \neg(\bar{x}\, R\, y) \vee \mathcal{P}(\bar{z}) \vee \neg(\bar{z}\, R\, f(y)) \vee \mathcal{P}(y) \vee \mathcal{P}(f(y)).$$

such that, additionally, if x is a variable occurring in a monadic atom $P(x)$ in C and if C contains a (negative) R literal then x occurs in at least one such R literal. We shall also write $C = C_y$ to emphasise the special role of y as the only variable that may occur on the right side of R literals in C, if there are any such literals. In that case, $C_{y'}$ will denote the clause in which y is replaced by y'.

In generalising from clauses of type (5) we have to consider the class of clauses:

(9) $$\mathcal{P}(\iota)$$

The clauses (2) and (7) are both instances of this slightly more general type of clauses:

(10) $$\mathcal{P}(x) \vee x\, R\, f(x)$$

In summary, the class $\mathcal{K}$ of clauses for which we want to show that saturation under C terminates consists of the clauses of type (6), (8), (9), and (10). Clearly, $\mathcal{K}$ contains all clauses that might result from the structural translation of modal formula, as well as the frame axioms we consider here.

The following theorem is true for finite signatures which we assume here.

Theorem 4. *$\mathcal{K}$ contains only finitely many condensed clauses (modulo variable renaming).*

Proof. Clearly there may be only finitely many condensed (hence fully factored) clauses of type (6), (9), or (10), as these clauses are flat, i.e. terms have at most height one, and contain at most one variable. So the only non-trivial case are the clauses of the type (8). These have the form

$$C_y = \mathcal{P}(\bar{x}) \vee \neg(\bar{x}\, R\, y) \vee \mathcal{P}(\bar{z}) \vee \neg(\bar{z}\, R\, f(y)) \vee \mathcal{P}(y) \vee \mathcal{P}(f(y))$$

in which we may view y as a global parameter or constant. Under this view, the R atoms $x_i\, R\, y$ and $z_j\, R\, f(y)$, respectively, play the role of monadic atoms $R_y(x_i)$ and $R_y^f(z_j)$. Essentially, C_y is a monadic clause, which may consist of exponentially many (in the cardinality of the signature) variable-disjoint subclauses, each of which contains one variable (besides y). A condensed clause of this type can be of at most exponential length. From this finiteness modulo variable renaming follows.

The proof shows that clauses of type (8) may be exponentially long in the size of the signature. That gives us a doubly-exponential space (and time) bound for our decision procedure. A more space-economic (single-exponential) representation would result from splitting the clauses C_y into their variable disjoint regions, connecting them with the help of auxiliary monadic predicates $A(y)$. The resulting clauses are again of the form (8) but have a linear length (in the size of the signature) only. Based on this splitting technique, a saturation-based decision procedure using C can be implemented in single-exponential time and space for any of the modal logics that can be translated into $\mathcal{K}$. Observe that condensation of the restricted type of clauses that we employ is an at most quadratic problem.

7 Closure under ordered chaining

For the clauses in $\mathcal{K}$ some of the inference schemes in C are obviously void. In particular, ordered chaining and the first variant of negative chaining cannot be applied. In fact, in any positive occurrence of R literals in either (6) or (10) the second argument is greater or equal than the first argument in any total reduction ordering. Similarly, transitivity resolution is void as there is no clause in $\mathcal{K}$ that has more than one positive R atom.

For demonstrating that $\mathcal{K}$ is closed also with respect to the remaining inferences of C we have to define an appropriate class of orderings and selection functions.

Let $\succ$ be any total reduction ordering on ground terms in which the constant ι is the minimal term. Let $>_{\mathbb{N}}$ be defined by $1 >_{\mathbb{N}} 0$. For every ground literal L, let

$$c_L = (\max_L, \mathrm{pol}_L, s_L)$$

where (i) $\max_L$ is the maximal argument of L with respect to $\succ$, (ii) pol_L is 1, if L is negative, and 0 otherwise, and (iii) s_L is 1, if L is a binary literal $(\neg)(s\, R\, t)$ and $s \succ t$, and 0 otherwise. The ordering $\succ_c$ on the complexity measure is then defined to be the lexicographic combination of $\succ$, $>_{\mathbb{N}}$, and $>_{\mathbb{N}}$.

For example, if $s \succ t$, then the complexity of $s\,R\,t$ is $(s, 0, 1)$, whereas the complexity of $\neg(t\,R\,s)$ is $(s, 1, 0)$. The maximal term is the main criterion. Observe also that 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 [3] so that the completeness theorem (Theorem 2) applies. Let us assume for the remainder of this paper that $\succ$ denotes one specific but arbitrary such ordering based on $\succ_c$.

The selection function S selects at most negative R literals in C. More specifically, if C contains a negative R literal of the form $\neg(s\,R\,t)$ such that $s \succeq t$, S selects one such literal. No other literals are selected by S.

We now proceed with the analysis of C inferences on clauses in $\mathcal{K}$, assuming $\succ$ and S as just specified.

Lemma 5. *Let C and D be clauses of the form (8). Any inference in* C *from premise(s) C (and D) produces a clause of the form (8).*

Proof. Since there are no positive occurrences of R in clauses of type (8), inferences by ordered chaining, negative chaining, and by ordered resolution and factoring with negative occurrences of R are not possible. Only inferences by ordered resolution which resolve a monadic atom need to be considered. Since there are no positive R literals in C or D, if one of these clauses has a negative R literal selected, the clause cannot participate in any inference. Hence we may subsequently assume that no literal is selected in C and D.

1. Suppose that the resolved atom in $C = C_y$ is the form $P(x)$. Consequently, the variable x represents the maximal term in C. In this case a term of the form $f(x)$ can not occur in C. x cannot occur as the first argument of an R literal as otherwise this R literal would be selected. Therefore x occurs as the right argument of an R atom, that is, $x = y$, if there are R literals in C.
   Let $\neg P(x')$ be the resolved literal in $D = D_{y'}$. By a similar reasoning we infer that $x' = y'$, if there are R literals in D, and that $f(x')$ does not occur in D. The conclusion of the resolution inference from C and D unifies y and y' and also satisfies the other properties of clauses of the form (8).

If $\neg P(f(x'))$ is the resolved literal in $D = D_{y'}$ then $x' = y'$, and the variable $x = y$ in C is replaced by $f(x')$. Thus, the result is again a clause of the form (8).

2. The remaining cases where $P(f(x))$ is strictly maximal in C, as well as the cases with dual polarity of the resolved atoms are symmetric to the first case.

The class of clauses in Lemma 5 remains closed under the ordered chaining calculus if we include reflexivity and certain ground clauses containing monadic predicates only.

Lemma 6. *Let C and D be clauses of the form (6), (8), or (9). Any inference in C from C and/or D produces a clause of the form (6), (8), or (9).*

Proof. By the Lemma 5, inferences from clauses of the form (8) produce clauses of the same type. Let us now consider inferences between clauses of the form (8) and (9). Let $C = C_y$ be a clause of the form (8) and let D be a clause of the form (9). The only inference possible is by ordered resolution. If $(\neg)P(x)$ is the resolved literal in C, we may again infer that $x = y$, if there are R literals in C, and that no term $f(x)$ occurs in C. Also, no R literal is selected in C. If C does not contain an R literal the result is a clause of the form (9). Suppose now that C does contain an R literal. In that case the variable x in C will be bound to the constant ι. Since ι is the minimal ground term but at the same time the maximal term in C, in the respective ground instance of C all variables are bound to ι. Thus one occurrence of an R literal in that instance of C must be selected. This is a contradiction.

Ordered resolution steps with the reflexivity clause (6) removes one occurrences of an R literal in the other premise. The result is a clause of the form (8).

Lemma 7. *Let C be a clause of the form (10) and let D be a clause of the form (6), (8), or (9). Any inference in C from the two premises C and D produces only clauses of the form (8).*

Proof. C is a clause of the form $C' \vee x' \, R \, f(x')$ where $x' \, R \, f(x')$ in C is maximal. Only if D is of the form (8) an inference is possible (by negative chaining or by ordered resolution).

1. In the case of ordered resolution a literal of the form $\neg(x \, R \, f(y))$ or of the form $\neg(x \, R \, y)$ is resolved in D. In the first case, this obviously results in a clause of the form (8). Suppose we resolve on a

literal $\neg(x\,R\,y)$ in D. We may assume that D does not contain any occurrences of a term $f(y)$, for otherwise, the literal $\neg(x\,R\,y)$ is not maximal. Since y is bound to the term $f(x)$ and x does not occur in other R literals in D the result is a clause of the form (8).
2. There are two cases of negative chaining on a maximal literal in D. The interesting case is again when $\neg(x\,R\,y)$ is maximal in D. We may assume by the ordering restrictions of negative chaining that $y \succ x$ and, as above, that D does not contain any occurrences of a term $f(y)$ for otherwise the literal $\neg(x\,R\,y)$ is not maximal. The difference to the situation above is that negative chaining corresponds to an attempt of a many-step rewrite proof. Thus $\neg(x\,R\,y)$ is not removed but rewritten to $\neg(x\,R\,x')$ where y is bound to $f(x')$, that is, all other occurrences of R literals in the resolvent are now of the form $\neg(u\,R\,f(x'))$. The result is a clause of the form (8).

Our main theorem now is this:

Theorem 8. *Let ϕ be a modal formula, let Σ be a possibly empty set of frame properties from Table 1, and let N be the set of clauses obtained by applying* $\mathrm{Cls}\Xi\Pi_r^{\Sigma}$ *to ϕ. Then,*

1. *any derivation from N in the ordered chaining calculus with eager condensation terminates, and*
2. *ϕ is unsatisfiable in $\mathsf{K}\Sigma$ if and only if the saturation of N under* C *contains the empty clause.*

Proof. From the Lemmas 5, 6, and 7 we may infer that the class $\mathcal{K}$ of clauses is closed under C inferences. In the Theorem 4 we have shown that $\mathcal{K}$ is finite if clauses are fully condensed. This shows the first part of the theorem. The second part is implied by the Theorem 1, together with the soundness and completeness of C, cf. Theorem 2.

This result extends to multi-modal logics with transitive modalities. The translation mapping π_r is then modified in the expected way. Each pair of modal operators $\Box_i$ and $\Diamond_i$ is associated with a distinguished binary relation symbol R_i:

$$\pi_r(\Box_i\phi, x) = \forall y\ (x\,R_i\,y \rightarrow \pi_r(\phi, y))$$
$$\pi_r(\Diamond_i\phi, x) = \exists y\ (x\,R_i\,y \wedge \pi_r(\phi, y)).$$

The generalisation of Theorem 8 is the following. $\bigoplus \mathsf{K}\Sigma_i$ denotes the fusion of a family of extensions of K [13]. Note, not all relations need to be transitive.

Theorem 9. *Let ϕ be a modal formula, let $\{\Sigma_i\}_i$ be a family of possibly empty sets of frame properties from Table 1, and let $\Sigma = \bigwedge \Sigma_i$. If $N = \mathrm{Cls}\Xi\Pi_r^{\Sigma}(\phi)$, then*

1. *any derivation from N in the ordered chaining calculus terminates, and*
2. *ϕ is unsatisfiable in $\bigoplus \mathsf{K}\Sigma_i$ if and only if N reduces to the empty clause in C.*

Proof. The general form of the clauses (8) needs to be adapted to:

$$\mathcal{P}(\bar{x}) \lor \neg(\bar{x}\,\mathcal{R}\,y) \lor \mathcal{P}(\bar{z}) \lor \neg(\bar{z}\,\mathcal{R}\,f(y)) \lor \mathcal{P}(y) \lor \mathcal{P}(f(y)),$$

where, similar as before, $\neg(\bar{x}_n\,\mathcal{R}\,t)$ expands to $\bigvee_i \neg(x_i\,\mathcal{R}\,t)$, $\neg(t\,\mathcal{R}\,\bar{x}_n)$ to $\bigvee_i \neg(t\,R\,x_i)$, and $\neg(s\,\mathcal{R}\,t)$ expands to $\neg(s\,R_1\,t) \lor \ldots \lor \neg(s\,R_m\,t)$. Because the modalities and the relations do not interact, the proof is essentially as for Theorem 8. For non-transitive relations no consideration of the application of the chaining rules is necessary.

Other frame properties can be embedded in our class of clauses. For example, $\forall x \exists y\, \neg(x\,R\,y)$ or $\forall x, y\, \neg(x\,R_i\,y) \lor \neg(x\,R_j\,y)$. We expect that the class of clauses can be extended to include also n-ary function symbols as in $\neg(x\,R_i\,y) \lor x\,R_j\,f(x,y)$, for example. It is also safe to allow more constants, in particular, ground clauses of the form $(\neg)P(a)$ or $(\neg)(a\,R\,b)$.

8 More examples

Reconsider the sample modal formula $\phi_1 = \Box\Diamond p$ from Section 4. The input clauses stemming from $\Pi_r^{\mathsf{T4}}(\phi_1)$ are

(11) xRx

(12) $Q_2(\iota)$

(13) $\neg Q_2(x) \lor \neg(x\,R\,y) \lor Q_1(y)$

(14) $\neg Q_1(x) \lor x\,R\,f_1(x)$

(15) $\neg Q_1(x) \lor P(f_1(x))$.

As in the derivation in Section 4, the inference step between clauses (13) and (14) on the R-literals is still possible in the ordered constraint chaining calculus, and results in

(16) $\neg Q_2(x) \lor Q_1(f_1(x)) \lor \neg Q_1(x)$.

But, an inference step by ordered resolution between clauses (16) and (14) is now impossible. The literal $\neg Q_1(f_1(x))$ is neither maximal nor is it selected in the instance $\neg Q_1(f_1(x)) \vee f_1(x)\, R\, f_1(f_1(x))$ of clause (14). Thus, the ordering restrictions prevent an resolution inference step on $Q_1(f_1(x))$ in clause (16). An inference step by ordered resolution between clauses (11) and (13) gives

(17) $\qquad \neg Q_2(x) \vee Q_1(x).$

Without loss of generality we assume that instances of $\neg Q_2(x)$ are $\succ$-maximal in this clause. Assuming that instances of $Q_1(x)$ are maximal would not affect termination. Then

(18) $\qquad \neg Q_1(\iota)$

is derived by resolving (17) and (12). Finally, it is possible to apply the negative chaining rule to (17) and (13), giving

(19) $\qquad \neg Q_2(x) \vee \neg(x\, R\, y) \vee \neg Q_1(y) \vee Q_1(f_1(y)).$

As with (16) and (14), further inference steps by negative chaining on (19) are prevented by the ordering restrictions. The clause set is now saturated. As it does not contain the empty clause, the clause set we have started from, and therefore also ϕ_1, are satisfiable.

Concerning the second example of Section 4, that is, the formula $\Box p$, not a single inference step is performed in the chaining calculus. The input clauses are:

(20) $\qquad Q_1(\iota)$

(21) $\qquad \neg Q_1(x) \vee \neg(x\, R\, y) \vee P(y)$

There are no positive R literals, and any inference upon $\neg Q_1(x)$ and $Q_1(\iota)$ is blocked by the selection function which selects $\neg(x\, R\, y)$.

We now consider a more complex example. The formula

$$\phi_2 = \Box(p_1 \vee p_2) \wedge \Diamond(\Box(\neg p_1 \vee p_2) \wedge \Diamond\Diamond\neg p_2)$$

is unsatisfiable in K4. The clausal form of $\Xi\Pi_r^{\mathsf{4}}(\phi_2)$ includes among others the following clauses.

(22) $\qquad \neg Q_1(x) \vee x\, R\, f_1(x)$

(23) $\qquad \neg Q_1(x) \vee \neg P_2(f_1(x))$

(24) $\qquad \neg Q_2(x) \vee x\, R\, f_2(x)$

(25) $\qquad \neg Q_2(x) \vee Q_1(f_2(x))$

(26) $\qquad \neg Q_3(x) \vee \neg P_1(x) \vee P_2(x)$

(27) $\qquad \neg Q_4(x) \vee \neg(x\, R\, y) \vee Q_3(y)$

(28) $\qquad \neg Q_5(x) \vee Q_2(x)$

(29)	$\neg Q_5(x) \vee Q_4(x)$
(30)	$\neg Q_6(x) \vee x\, R\, f_3(x)$
(31)	$\neg Q_6(x) \vee Q_5(f_3(x))$
(32)	$\neg Q_7(x) \vee P_1(x) \vee P_2(x)$
(33)	$\neg Q_8(x) \vee \neg(x\, R\, y) \vee Q_7(y)$
(34)	$\neg Q_9(x) \vee Q_6(x)$
(35)	$\neg Q_9(x) \vee Q_8(x)$
(36)	$Q_9(\iota)$

Note that $Q_8(x)$ can be interpreted as '$\Box(p_1 \vee p_2)$ holds at world x'. The literals $Q_4(x)$, $Q_2(x)$, and $Q_2(x)$ have an analogous meaning for the subformulae $\Box(\neg p_1 \vee p_2)$, $\Diamond\Diamond\neg p_2$, and $\Diamond\neg p_2$, respectively. In the following derivation condensing steps are not explicitly stated. By [(1)2,R,(2)1] we denote that the second literal of the first clause is resolved with the first literal of the second clause. Analogously, [(1)2,NC,(2)1] denotes an inference by negative chaining.

[(26)2,R, (32)2]	(37)	$\neg Q_3(x) \vee \neg Q_7(x) \vee P_2(x)$
[(37)3,R, (23)2]	(38)	$\neg Q_3(f_1(y)) \vee \neg Q_7(f_1(y)) \vee \neg Q_1(y)$
[(27)2,NC,(22)2]	(39)	$\neg Q_4(x) \vee \neg(x\, R\, y) \vee \neg Q_1(y) \vee Q_3(f_1(y))$
[(33)2,NC,(22)2]	(40)	$\neg Q_8(x) \vee \neg(x\, R\, y) \vee \neg Q_1(y) \vee Q_7(f_1(y))$
[(38)1,R, (39)4]	(41)	$\neg Q_4(x) \vee \neg(x\, R\, y) \vee \neg Q_1(y) \vee \neg Q_7(f_1(y))$
[(41)4,R, (40)4]	(42)	$\neg Q_4(x) \vee \neg(x\, R\, y) \vee$ $\neg Q_8(z) \vee \neg(z\, R\, y) \vee \neg Q_1(y)$

Clause (42) is interesting. It says that if $\Box(\neg p_1 \vee p_2)$ holds at a world x, $\Box(p_1 \vee p_2)$ holds at world z, and there is a world y which is accessible from both x and z, then $\neg\Diamond\neg p_2$, that is $\Box p_2$, holds in y. No assumptions are made as to whether x is accessible from z or vice versa. Note that this property cannot be expressed without the object language containing explicit representations of (universally quantified) worlds and the accessibility relation. This is one of the major factors which enables us to maintain all the information which needs to be derived in the restricted form of (8). The remainder of the refutation is as follows.

[(42)2,R, (24)2]	(43)	$\neg Q_4(x) \vee \neg Q_2(x) \vee \neg Q_8(z) \vee$ $\neg(z\, R\, f_2(x)) \vee \neg Q_1(f_2(x))$
[(43)4,NC,(24)2]	(44)	$\neg Q_4(x) \vee \neg Q_2(x) \vee \neg Q_8(z) \vee$ $\neg(z\, R\, x) \vee \neg Q_1(f_2(x))$

[(44)5,R,	(25)2]	(45)	$\neg Q_4(x) \vee \neg Q_2(x) \vee \neg Q_8(z) \vee \neg(z\,R\,x)$
[(45)4,R,	(30)2]	(46)	$\neg Q_4(f_3(x)) \vee \neg Q_2(f_3(x)) \vee \neg Q_8(x) \vee \neg Q_6(x)$
[(46)1,R,	(28)2]	(47)	$\neg Q_4(f_3(x)) \vee \neg Q_5(f_3(x)) \vee \neg Q_8(x) \vee \neg Q_6(x)$
[(47)1,R,	(31)2]	(48)	$\neg Q_4(f_3(x)) \vee \neg Q_8(x) \vee \neg Q_6(x)$
[(48)1,R,	(29)2]	(49)	$\neg Q_5(f_3(x)) \vee \neg Q_8(x) \vee \neg Q_6(x)$
[(49)1,R,	(31)2]	(50)	$\neg Q_8(x) \vee \neg Q_6(x)$
[(50)2,R,	(34)2]	(51)	$\neg Q_8(x) \vee \neg Q_9(x)$
[(51)1,R,	(35)2]	(52)	$\neg Q_9(x)$
[(52)1,R,	(36)1]	(53)	$\bot$

9 Further work

The approach purported in this paper is that modal logics are fragments of first-order logic, a view which has stimulated the work on the guarded fragment [1,6]. Although the guarded fragment is a generalisation of basic modal logic and includes also properties of the accessibility relation, like reflexivity and symmetry, transitivity is not within the scope of this fragment. Transitivity, however, has been our primary interest here. Since de Nivelle [6] has recently shown that a decision procedure for the guarded fragment based on an ordering refinement exists, it would be interesting to investigate the combination with chaining to obtain practical decidability results for an even broader generalisation of modal logics.

References

1. H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. Research Report ML-1996-03, Institute for Logic, Language and Computation, University of Amsterdam, The Netherlands, 1996.
2. M. Baaz, C. Fermüller, and A. Leitsch. A non-elementary speed-up in proof length by structural clause form transformation. In *Proc. of the Ninth Annual IEEE Symposium on Logic in Computer Science (LICS'94)*, pages 213–219. IEEE Computer Society Press, 1994.
3. L. Bachmair and H. Ganzinger. Ordered chaining calculi for first-order theories of binary relations. Research report MPI-I-95-2-009, Max-Planck-Institut für Informatik, Saarbrücken, Germany, October 1995. To appear in the *Journal of the ACM*.
4. L. Bachmair and H. Ganzinger. A theory of resolution. Research report MPI-I-97-2-005, Max-Planck-Institut für Informatik, Saarbrücken, Germany, April 1997. Preliminary version of a chapter in the *Handbook of Automated Reasoning*, edited by J. A. Robinson and A. Voronkov.
5. T. Boy de la Tour. An optimality result for clause form translation. *Journal of Symbolic Computation*, 14:283–301, 1992.

6. H. de Nivelle. A resolution decision procedure for the guarded fragment. In *Proc. of the 15th Int. Conf. on Automated Deduction (CADE-15)*, LNAI 1421, pages 191–204. Springer, 1998.
7. C. Fermüller, A. Leitsch, T. Tammet, and N. Zamov. *Resolution Method for the Decicion Problem*. LNCS 679. Springer, 1993.
8. U. Hustadt and R. A. Schmidt. An empirical analysis of modal theorem provers. To appear in the *Journal of Applied Non-Classical Logics*.
9. U. Hustadt and R. A. Schmidt. On evaluating decision procedures for modal logic. In *Proc. of the Fifteenth Int. Joint Conf. on Artificial Intelligence (IJCAI'97)*, pages 202–207. Morgan Kaufmann, 1997.
10. U. Hustadt and R. A. Schmidt. Issues of decidability for description logics in the framework of resolution. Submitted for publication, September 1998.
11. U. Hustadt, R. A. Schmidt, and C. Weidenbach. Optimised functional translation and resolution. In *Proc. of the Int. Conf. on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX'98)*, LNAI 1397, pages 36–37. Springer, 1998.
12. W. H. Joyner Jr. Resolution strategies as decision procedures. *Journal of the ACM*, 23(3):398–417, July 1976.
13. M. Kracht and F. Wolter. Properties of independently axiomatizable bimodal logics. *Journal of Symbolic Logic*, 56(4):1469–1458, 1991.
14. A. Nonnengart. Strong skolemization. Research Report MPI-I-96-2-010, Max-Planck-Institut für Informatik, Saarbrücken, 1996.
15. A. Nonnengart, G. Rock, and C. Weidenbach. On generating small clause normal forms. In *Proc. of the 15th Int. Conf. on Automated Deduction (CADE-15)*, LNAI 1421, pages 397–411. Springer, 1998.
16. R. A. Schmidt. Decidability by resolution for propositional modal logics. To appear in *Journal of Automated Reasoning*.
17. R. A. Schmidt. *Optimised Modal Translation and Resolution*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 1997.
18. R. A. Schmidt. Resolution is a decision procedure for many propositional modal logics. In *Advances in Modal Logic, Volume 1*, Lecture Notes 87, pages 189–208. CSLI Publications, Stanford, 1998.
19. G. S. Tseitin. On the complexity of derivations in propositional calculus. In *Studies in Constructive Mathematics and Mathematical Logic, Part II*, pages 115–125. Consultants Bureau, New York, 1970.