Developing Modal Tableaux and Resolution Methods via First-Order Resolution

RENATE A. SCHMIDT

ABSTRACT. This paper explores the development of calculi using different proof methods for propositional dynamic modal logics. Dynamic modal logics are *PDL*-like extended modal logics which are closely related to description logics. We show how tableau systems, modal resolution systems and Rasiowa-Sikorski systems, which are dual tableau systems, can be developed and studied by using standard principles and methods of first-order theorem proving. We translate modal logic reasoning problems to first-order clausal form and then use a suitable refinement of resolution to construct and mimic derivations using the desired proof method. The inference rules of the calculus can then be read off from the clausal form used.

Keywords: Modal logic, deduction, resolution, tableaux, dual resolution, dual tableaux, modal resolution, decidability.

1 Introduction

In this paper we discuss and extend an approach of developing tableaux calculi for modal logics that has been suggested and followed in our previous work [8, 13, 15, 16, 18, 32]. Although resolution calculi apparently operate considerably differently from tableau calculi, we have shown that it is possible to linearly simulate many forms of modal logic or description logic tableau calculi with standard techniques in resolution-based theorem proving. In [16] we have shown how derivations and search in standard tableau algorithms of the description logic $\mathcal{ALC}$ can be linearly simulated by resolution. This corresponds to local satisfiability testing in the basic multi-modal logic $K_{(m)}$. Using redundancy elimination techniques and a blocking rule we have shown in [15] how to mimic standard tableau algorithms for $\mathcal{ALC}$ with respect to non-empty TBoxes. This corresponds to local satisfiability in multi-modal logic $K_{(m)}$ with respect to a background theory of modal logic formulae (i.e. a set of non-logical axioms). In [18] details can be found of how to simulate derivations in the prefixed single-step tableau calculi of Massacci [20]. These simulation results show that it is possible to use first-order resolution in a way that it simulates modal logic and description logic tableau procedures. The close connection exhibited in these papers between tableau and a certain instance of resolution is exploited in [8] in order to develop a tableau calculus for a logic that has not been considered before. The logic considered was the dynamic modal logic $K_{(m)}(\wedge, \vee, \smile)$. Dynamic modal logics are *PDL*-like modal logics in which the parameters of the modal operators can be relational formulae, which are interpreted as actions or programs in *PDL*, and are closely related to description logics [33]. $K_{(m)}(\wedge, \vee, \smile)$ is the multi-modal logic defined over frames in which the relations are closed under intersection, union and converse. The logic corresponds to the description logic $\mathcal{ALC}$ in which conjunction, disjunction and converse of roles are allowed. In [8] we have shown how a tableau calculus can essentially be 'read off' from the clausal form of the translation of formulae in $K_{(m)}(\wedge, \vee, \smile)$.

It is this *'develop via first-order resolution'* approach which we explore and extend in this paper. We consider in more detail how tableau calculi can be developed for modal logics via a suitable translation to first-order logic and resolution. However we also show that the approach can be used to develop other kinds of deduction methods. In particular, we show how the approach can be extended to develop Rasiowa-Sikorski systems. These are tableau-style calculi for testing the validity of formulae [25]. In addition, we consider the development of modal resolution systems which operate directly on modal logic formulae.

We show that all three types of calculi (tableau, Rasiowa-Sikorski, modal resolution) can be obtained naturally via translation to first-order logic and standard techniques of resolution theorem proving. Key to the 'develop via first-order resolution' approach are two aspects: (i) An effective, sound and complete reduction to first-order logic that retains enough information about the input formula in the source logic so that the inference rules can simply be read off from the clausal form. (ii) A refinement of first-order resolution which performs inferences exactly like the kind of system we want to develop.

The form and property of the calculus one obtains depends very much on the reduction and the refinement used. Small modifications result in different styles and variations of calculi. In this paper we focus on the development of *ground semantic calculi*. By this we mean calculi which operate on labelled modal formulae. For each operator in the logic there is a decomposition rule which basically 'breaks down' formulae into less complex formulae on the basis of the semantics of the top-level operator in one of the premises. The labels are given by constants (or ground Skolem terms) which represent states in the underlying Kripke model. Currently, ground semantic tableau calculi appear to be the preferred style of tableau calculus used in the area, and many modal and description logic theorem provers are based on ground semantic calculi.

The cited previous work shows that ground semantic tableau calculi can be simulated by a structural reduction into range-restricted clauses in combination with hyperresolution. A clause is range-restricted if all variables of the clause occur in the negative literals of that clause. Hyperresolution on range-restricted clauses has the property that all positive premises are ground clauses and all conclusions are ground clauses. This is precisely the property which, when using a structural reduction, allows us to interpret the negative, non-ground premise as an inference rule $\mathcal{I}$ of the ground calculus. The positive, ground premises of a hyperresolution inference step represent then the premises of the rule $\mathcal{I}$, and the conclusions represent the conclusions of the rule. Combined with splitting, hyperresolution allows us to simulate and develop ground semantic tableau calculi. This is shown in Section 4. In Section 5 we discuss how dual hyperresolution with splitting allows us to simulate and develop ground semantic Rasiowa-Sikorski calculi. Furthermore, hyperresolution without a splitting rule produces modal resolution calculi, as is discussed in Section 6.

To illustrate the approach we focus on the development of calculi for the dynamic modal logic $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$, i.e. multi-modal logic defined over relations closed under intersection, union, converse and domain restriction. As far as we know this logic has not been considered before and certainly no tableau calculus has been described for it in the literature. $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$ is subsumed by Peirce logic, for which tableau calculi are defined in [22, 34]. However, the relational disjunction operator and the domain restriction operator of $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$ are not explicit operators in Peirce logic, so there are no tableau rules for relational disjunction and domain restriction in the existing tableau calculi for Peirce logic. Sound rules can be easily defined for these operators, but, on the one hand, the logic lacks the symmetry that Peirce logic has because of the absence of relational negation in $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$. On the other hand, since $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$ is decidable and Peirce logic is not decidable, $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$ is a good candidate for exploring the possibilities of developing special-purpose decision procedures for it via the approach we are purporting in this paper.

In the next section we recall standard definitions of resolution-based first-order theorem proving. Section 3 introduces dynamic modal logics and $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$. It also defines the structural translation and reduction to clausal form that we use. The main parts are Sections 4–6, in which we describe, in turn, how ground semantic tableau calculi, ground semantic Rasiowa-Sikorski calculi, and ground modal resolution calculi can be developed in a systematic way via first-order resolution. The final section discusses the significance and consequences of the method, and mentions future work.

Throughout the paper we use the notation and terminology of our previous papers, cf. e.g. the surveys [31, 33].

2 First-Order Resolution

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.

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

This says that any first-order formula can be transformed efficiently into a satisfiability equivalent set of clauses. 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 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 contraction rule, i.e. it 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 provide a sound and refutationally complete calculus for propositional logic and sets of ground clauses. We obtain a sound and refutationally complete inference system for full first-order logic and clause logic, if we augment the rules with unification. This calculus, the *basic resolution calculus* [29], is sound and complete for full first-order logic and clause logic. It is however very prolific in generating new clauses. This was noticed already in the very early days of the development of first-order resolution methods. The first papers, by Robinson and others, 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 [36], E [35] and VAMPIRE [27] (in order of creation), are based on the modern framework of saturation-based resolution and superposition. In the following, when we refer to *resolution* we mean this framework (cf. e.g. [3, 23]).

The main ingredients of the 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 refutation proof can be found more quickly as the search space for the proof is reduced. There is a general completeness proof which requires only weak conditions for the admissibility of orderings and selection functions [3].

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. Standard simplification rules like elimination of duplicate literals within a clause, tau-

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

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

Simplify: $\dfrac{N}{(N \backslash M) \cup M'}$ 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: $\dfrac{N \uplus \{C \vee D\}}{N \cup \{C\} \mid N \cup \{D\}}$ if C and D are variable-disjoint.

Resolvents and factors are computed with:

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

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. The left (right) premise is called the *positive (negative) premise*.

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

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 $C\sigma$.

Figure 1. The resolution calculus $R^{\mathrm{red}}_{\mathrm{sp}}$.

tology deletion, subsumption deletion (forward and backward subsumption deletion), condensing, etc, are instances of this notion [3].

Let $R^{\mathrm{red}}_{\mathrm{sp}}$ 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'. $\uplus$ denotes disjoint union.) 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 [3]. 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. A non-ground clause is redundant in N if all its ground instances are redundant in N. 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 full generality, instead only effectively computable instances of the Delete and Simplify rules are implemented.

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 $N \cup \{D\}$ (alternatively, it is possible to use the complement splitting rule, which

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. The rightmost premise in the rule is referred to as the *negative premise* and all other premises are referred to as *positive premises*.

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

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 $C\sigma$.

Figure 2. The Deduce rules of ordered hyperresolution.

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 an alternative to explicit splitting is splitting through new propositional variables [7, 26] implemented in the theorem prover VAMPIRE [27] or the generalisation called separation in [30].

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

THEOREM 2 (Bachmair et al [3, 4]). *$\mathsf{R}^{\mathrm{red}}_{\mathrm{sp}}$, $\mathsf{R}^{\mathrm{red}}$ and R are sound and complete refutation systems 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 hyperresolution rule, positive factoring, redundancy elimination and splitting. Similar as above, $\mathsf{OH}^{\mathrm{red}}$ denotes the calculus $\mathsf{OH}^{\mathrm{red}}_{\mathrm{sp}}$ but without the splitting rule, OH denotes the calculus just consisting of Deduce rules, and $\mathsf{OH}_{\mathrm{sp}}$ is OH with splitting. For completeness an ordering refinement is optional. We use the notation $\mathsf{H}^{\mathrm{red}}_{\mathrm{sp}}$, $\mathsf{H}^{\mathrm{red}}$, H and H_{sp} for the unordered versions.

COROLLARY 3. *$\mathsf{OH}^{\mathrm{red}}_{\mathrm{sp}}$, $\mathsf{OH}^{\mathrm{red}}$, $\mathsf{OH}_{\mathrm{sp}}$, OH, $\mathsf{H}^{\mathrm{red}}_{\mathrm{sp}}$, $\mathsf{H}^{\mathrm{red}}$, H_{sp} and H are sound and complete refutation systems for clause sets.*

3 Dynamic Modal Logics

A dynamic modal logic is an extension of the multi-modal logic $K_{(m)}$ in which the modal operators are parameterised by relational formulae [33].

Given countably many propositional variables denoted by p_j, 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 $\alpha \wedge \beta$, $\alpha \vee \beta$, $\alpha^{\smile}$ and $\alpha\upharpoonright\phi$ are relational formulae. In $K_{(m)}$ the only relational formulae are relational variables.

Thus the language of dynamic modal logics consists of two syntactic types: dynamic modal formulae and relational formulae. The logical connectives are (i) the connectives of the basic multi-modal logic $K_{(m)}$, with the difference that the modal operators are indexed with relational formulae, and (ii) a finite set of relational operators. A dynamic modal logic with relational operators $\star_1, \ldots, \star_k$ is denoted by $K_{(m)}(\star_1, \ldots, \star_k)$.

The semantics of a dynamic modal logic is defined in terms of frames. 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. A model is 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 conditions (R_α is the preferred notation for $R(\alpha)$):

$$\begin{aligned}
&\mathcal{M}, x \models p \text{ iff } x \in v(p), \qquad\qquad \mathcal{M}, x \models \neg\phi \text{ iff } \mathcal{M}, x \not\models \phi,\\
&\mathcal{M}, x \models \phi \wedge \psi \text{ iff both } \mathcal{M}, x \models \phi \text{ and } \mathcal{M}, x \models \psi,\\
&\mathcal{M}, x \models [\alpha]\phi \text{ iff } (x, y) \in R_\alpha \text{ implies } \mathcal{M}, y \models \phi, \text{ for any } y \in W.
\end{aligned}$$

In addition, the following conditions are satisfied:

$$\begin{aligned}
R_{\alpha\wedge\beta} &= R_\alpha \cap R_\beta, & R_{\alpha\vee\beta} &= R_\alpha \cup R_\beta,\\
R_{\alpha^{\smile}} &= R_\alpha^{\smile}, & R_{\alpha\upharpoonright\phi} &= \{(x, y) \mid (x, y) \in R_\alpha \wedge \mathcal{M}, x \models \phi\}.
\end{aligned}$$

$R^{\smile}$ denotes the converse (or inverse) of a relation R. We can define the range restriction operator by $\alpha\downharpoonleft\phi =^{\text{def}} (\alpha^{\smile}\upharpoonright\phi)^{\smile}$.

If $\mathcal{M}, x \models \varphi$ holds then φ is *(locally) true* at x in $\mathcal{M}$ and $\mathcal{M}$ *(locally) satisfies* φ. A modal formula φ is *(locally) 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 *(locally) valid* iff it is (locally) satisfiable in every world of all models.

The following result is a consequence of decidability results in [16] and also of the decidability of the two-variable fragment of first-order logic.

THEOREM 4. *Let L be a dynamic modal logic with any subset of* $\{\wedge, \vee, {}^{\smile}, \upharpoonright, \downharpoonleft\}$ *as relational operators. The local (and global) satisfiability problem in L is decidable.*

We know that the result remains true when we allow negation as a relational operator [16], or relational composition and identity [21]. However, any dynamic modal logic with relational conjunction, relational negation and (negative occurrences of) composition is undecidable [31, 33].

Structural Transformation into Clausal Form

For efficiency reasons and in order to be able to have better control over the resolution inferences performed on the clausal form, we use a *structural* *version of the relational translation* of dynamic modal logics into first-order logic. The translation is similar to the one used in [32]; other structural translations have been used in e.g. [8, 15, 16, 18].

Throughout the paper, we assume that all occurrences of double negation are eliminated from modal formulae and conjunction is an idempotent operator. For any formula F, let $\sim F$ denote G if $F = \neg G$, and $\neg F$ otherwise. Let Def be a transformation of dynamic modal formulae and relational formulae defined as follows.

$$\begin{aligned}
\mathrm{Def}(\psi) =^{\mathrm{def}}\ & \forall x\,(Q_\psi(x) \rightarrow \pi(\psi, x)) \wedge \forall x\,(Q_{\sim\psi}(x) \rightarrow \pi(\sim\psi, x)) \\
& \wedge \forall x\,(Q_\psi(x) \rightarrow \neg Q_{\sim\psi}(x)), \\
\mathrm{Def}(\alpha) =^{\mathrm{def}}\ & \forall x\,y\,(R_\alpha(x,y) \rightarrow \pi(\alpha, x, y)) \wedge \forall x\,y\,(\pi'(\alpha, x, y) \rightarrow R_\alpha(x,y)).
\end{aligned}$$

$\mathrm{Def}(\psi)$ is the *definition* of Q_ψ, which is a new predicate symbol uniquely associated with the modal formula ψ. Similarly, $\mathrm{Def}(\alpha)$ is the definition of the new symbol R_α uniquely associated with the relational formula α. π is defined as follows (z is any variable distinct from x):

$$\begin{aligned}
\pi(p, x) &= \top, \\
\pi(\neg p, x) &= \neg Q_p(x), \\
\pi(\psi \wedge \phi, x) &= Q_\psi(x) \wedge Q_\phi(x), \\
\pi(\neg(\psi \wedge \phi), x) &= Q_{\sim\psi}(x) \vee Q_{\sim\phi}(x), \\
\pi([\alpha]\psi, x) &= \forall z\,(R_\alpha(x,z) \rightarrow Q_\psi(z)), \\
\pi(\neg[\alpha]\psi, x) &= \exists z\,(R_\alpha(x,z) \wedge Q_{\sim\psi}(z)), \\
\pi(r, x, y) &= \pi'(r, x, y) = R_r(x,y), \\
\pi(\alpha \wedge \beta, x, y) &= \pi'(\alpha \wedge \beta, x, y) = R_\alpha(x,y) \wedge R_\beta(x,y), \\
\pi(\alpha \vee \beta, x, y) &= \pi'(\alpha \vee \beta, x, y) = R_\alpha(x,y) \vee R_\beta(x,y), \\
\pi(\alpha^{\smile}, x, y) &= \pi'(\alpha^{\smile}, x, y) = R_\alpha(y,x), \\
\pi(\alpha\upharpoonright\phi, x, y) &= R_\alpha(x,y) \wedge Q_\phi(x), \\
\pi'(\alpha\upharpoonright\phi, x, y) &= R_\alpha(x,y) \wedge \neg Q_{\sim\phi}(x).
\end{aligned}$$

Let $\mathrm{Def}(X) =^{\mathrm{def}} \{\mathrm{Def}(F) \mid F \in X\}$, if X denotes a set of modal and relational formulae. By $\mathrm{Sf}(F)$ we denote the set of all modal and relational subformulae of F.

THEOREM 5. *Let L be a dynamic modal logic defined over the operators $\{\wedge, \vee, {}^{\smile}, \upharpoonright\}$, and let φ be any modal formula. Suppose N is the set of clauses obtained from $\varphi' =^{\mathrm{def}} \exists x\, Q_\varphi(x) \wedge \bigwedge \mathrm{Def}(\mathrm{Sf}(\varphi))$ by transformation into conjunctive normal form, inner Skolemisation, and clausifying the Skolemised formula. Then: (i) Each clause in N is either a unit clause $Q(a)$, for some Skolem constant a, or it is an instance of a definitional clause given in Figure 3. (ii) φ is locally satisfiable in L iff φ' is first-order satisfiable iff N is first-order satisfiable.*

Note that the set N is computable in linear time, but may contain some obvious redundancies. These can be avoided by more careful introduction

Subformula θ	Definitional clauses for Def(θ)
(shortcut)	$\neg Q_{\neg\psi}(x)^+ \vee \neg Q_{\psi}(x)^+$
$\psi \wedge \phi$	$\neg Q_{\psi\wedge\phi}(x)^+ \vee Q_{\psi}(x)$
	$\neg Q_{\psi\wedge\phi}(x)^+ \vee Q_{\phi}(x)$
$\neg(\psi \wedge \phi)$	$\neg Q_{\neg(\psi\wedge\phi)}(x)^+ \vee Q_{\sim\psi}(x) \vee Q_{\sim\phi}(x)$
$[\alpha]\psi$	$\neg Q_{[\alpha]\psi}(x)^+ \vee \neg R_{\alpha}(x,y)^+ \vee Q_{\psi}(y)$
$\neg[\alpha]\psi$	$\neg Q_{\neg[\alpha]\psi}(x)^+ \vee R_{\alpha}(x, f_{\neg[\alpha]\psi}(x))$
	$\neg Q_{\neg[\alpha]\psi}(x)^+ \vee Q_{\sim\psi}(f_{\neg[\alpha]\psi}(x))$
$\alpha \wedge \beta$	$\neg R_{\alpha\wedge\beta}(x,y)^+ \vee R_{\alpha}(x,y)$
	$\neg R_{\alpha\wedge\beta}(x,y)^+ \vee R_{\beta}(x,y)$
	$R_{\alpha\wedge\beta}(x,y) \vee \neg R_{\alpha}(x,y)^+ \vee \neg R_{\beta}(x,y)^+$
$\alpha \vee \beta$	$\neg R_{\alpha\vee\beta}(x,y)^+ \vee R_{\alpha}(x,y) \vee R_{\beta}(x,y)$
	$R_{\alpha\vee\beta}(x,y) \vee \neg R_{\alpha}(x,y)^+$
	$R_{\alpha\vee\beta}(x,y) \vee \neg R_{\beta}(x,y)^+$
$\alpha^{\smile}$	$\neg R_{\alpha^{\smile}}(x,y)^+ \vee R_{\alpha}(y,x)$
	$R_{\alpha^{\smile}}(x,y) \vee \neg R_{\alpha}(y,x)^+$
$\alpha\restriction\phi$	$\neg R_{\alpha\restriction\phi}(x,y)^+ \vee R_{\alpha}(x,y)$
	$\neg R_{\alpha\restriction\phi}(x,y)^+ \vee Q_{\phi}(x)$
	$R_{\alpha\restriction\phi}(x,y) \vee \neg R_{\alpha}(x,y)^+ \vee Q_{\sim\phi}(x)$

Figure 3. Definitional clausal forms for $K_{(m)}(\wedge, \vee, ^{\smile}, \restriction)$.

of new symbols using a polynomial time algorithm. For efficiency reasons it is also sensible to take the polarities of all the occurrences of a subformula in the input problem into account in the specification of Def.

4 Tableau Systems

Numerous tableau methods have been developed, studied and also implemented for traditional-style modal logics, c.f. e.g. [6, 14, 20]. Tableau methods for some dynamic modal logics or logics equivalent to dynamic modal logics can be found in the description logic literature, e.g. [2, 21] and also here [8, 22, 34].

Simulating tableau systems for $K_{(m)}$

In this section, let us first look at how resolution can simulate ground semantic tableau for local satisfiability in the basic multi-modal logic $K_{(m)}$.

For this we define the notions of step-wise simulation, p-simulation and bisimulation. Let L_1 and L_2 be two logics and suppose Π is a sound and complete translation of (sets of) formulae in L_1 to (sets of) formulae in L_2, i.e. $\models N$ iff $\models \Pi(N)$ for any set N of formulae in L_1. In addition we assume that Π is computable in linear or polynomial time. Let $\mathcal{C}_1$ be a calculus, or proof procedure, for L_1, and let $\mathcal{C}_2$ be a calculus, or proof procedure, for L_2. By definition, $\mathcal{C}_2$ *step-wise simulates* $\mathcal{C}_1$ (*with respect to* Π) iff there

$$(\bot)\ \frac{s:\psi,\, s:\neg\psi}{\bot}$$

$$(\wedge)_1\ \frac{s:\psi\wedge\phi}{s:\psi} \qquad (\wedge)_2\ \frac{s:\psi\wedge\phi}{s:\phi} \qquad (\neg\wedge)\ \frac{s:\neg(\psi\wedge\phi)}{s:\sim\psi \mid s:\sim\phi}$$

$$(\neg\Box_i)_1\ \frac{s:\neg\Box_i\psi}{(s,t):R_i} \qquad (\neg\Box_i)_2\ \frac{s:\neg\Box_i\psi}{t:\sim\psi} \qquad (\Box_i)\ \frac{s:\Box_i\psi,\,(s,t):R_i}{t:\psi}$$

The side conditions of the $(\neg\Box_i)_j$ rules ($j \in \{1,2\}$) are that t is a constant uniquely associated with the premise $s:\neg\Box_i\psi$.

Figure 4. Tableau calculus for $K_{(m)}$.

is an n such that for every inference step in $\mathcal{C}_1$ there are at most n inference steps in $\mathcal{C}_2$ which derive the corresponding conclusion. More precisely, if N_i and N_{i+1} are consecutive sets in any $\mathcal{C}_1$-derivation then $\Pi(N_{i+1})$ can be obtained by at most n inference steps in $\mathcal{C}_2$ from $\Pi(N_i)$. If rather than '$\models N$ iff $\models \Pi(N)$' we have that '$\models N$ iff $\neg\Pi(N)$ is unsatisfiable', then we say $\mathcal{C}_2$ *step-wise simulates* $\mathcal{C}_1$ (*with respect to* Π) *in a dual manner*. We say that $\mathcal{C}_2$ *p-simulates* (proofs in) $\mathcal{C}_1$ iff the following condition holds: There is a function g computable in polynomial time which maps any proof (or refutation) in $\mathcal{C}_1$ to a proof (or refutation) in $\mathcal{C}_2$. If two calculi p-simulate each other, we say there is a *bisimulation* between them.

A ground semantic tableau calculus for $K_{(m)}$ is given by the rules in Figure 4 (c.f. e.g. [9, 14]). Definitions in the literature often include only one rule for conjunctions and negated box formulae, but for reasons which become obvious once we look at the simulation by resolution we choose the definitions given.

A *tableau derivation* is a finitely branching tree whose nodes are sets of labelled formulae. Given that φ is a formula to be tested for local satisfiability the root node is the set $\{a:\varphi\}$, where a denotes a constant. Successor nodes are constructed in accordance with a set of expansion rules. Expansion rules have the general form $X \,/\, X_1|\ldots|X_n$, where X is the set of premises and the X_i are sets of conclusions. An expansion rule is applicable to a selected labelled formula F in a node of the tableau if F, together with other formulae in the node, are simultaneous instantiations of all the premises of the rule. n successor nodes are created which contain the formulae of the current node and the appropriate instances of X_i. As usual it is assumed that in a derivation no rule is applied twice to the same set of instances of premises of a rule.

Now consider the clausal form of the structural translation of φ as defined in Theorem 5 (Figure 3). In particular, suppose N is the clausal form of $\exists x\, Q_\varphi(x) \wedge \bigwedge \mathrm{Def}(\mathrm{Sf}(\varphi))$. The definitional clauses relevant for $K_{(m)}$ are in the top-half of the table. Notice that for each of these definitional clauses there is one corresponding tableau expansion rule in Figure 4, and vice versa. The connection is the following: Every application of a tableau rule can be simulated by a hyperresolution inference step, or by a hyperresolution inference step followed immediately by splitting. If φ is the input formula, the derivation in hyperresolution starts with an inference with the (only positive) clause $Q_\varphi(a)$. This corresponds to the root of the tableau derivation given by $\{a : \varphi\}$. Subsequent inference steps are hyperresolution inference steps with definitional clauses as negative premises, possibly combined with splitting steps.

Let H_{sp} denote the unordered hyperresolution calculus with splitting. Further, assume that deletion rules, splitting, and the deduction rules are applied in this order, except that splitting is not applied to clauses which contain a selected literal.

LEMMA 6 ([8]). *H_{sp} on the structural transformation defined in Theorem 5 step-wise simulates ground labelled tableau for local satisfiability in $K_{(m)}$.*

THEOREM 7 ([8]). *There is a linear bisimulation between the tableau calculus in Figure 4 and H_{sp} on the structural transformation for local satisfiability in $K_{(m)}$.*

In fact, not only can corresponding proofs be constructed in a systematic manner, also, corresponding derivations can be constructed in the same way (i.e. even derivations that are not proofs). In addition the overhead is negligible. Hence the search complexity of the tableau system and H_{sp} are identical. Similar simulation results have been shown in [18] for prefixed single-step tableau calculi of K, and K extended with the axioms D, T and B. (The results are also true for multi-modal $K_{(m)}$ with D, T and B modalities.) Using the axiomatic translation method [32] the results can be strengthened to prefixed single-step tableau calculi of other traditional modal logics.

Developing tableau systems for $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonleft)$

Let us now illustrate how the principles of the 'develop via first-order resolution' approach can be applied to the dynamic modal logic $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonleft)$.

We can prove the following:

THEOREM 8. *Let φ be any $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonleft)$-formula and let N be the clausal form of the structural transformation defined in Theorem 5. Then: (i) Any H_{sp}-derivation from N terminates. (ii) φ is locally unsatisfiable in $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonleft)$ iff the H_{sp}-saturation of N contains the empty clause.*

Part (i) can be shown using an argument similar to the proof of Theorem 6.6 in [18] and Theorem 7.7 in [8]. Part (ii) is a consequence of Theorem 5 and Corollary 3.

The definitional clauses of the input set have the form as specified in Figure 3. The literals selected by the selection function of the calculus H_{sp} are marked with $^{+}$. The only other clause in any input set is a ground unit clause of the form $Q_\varphi(a)$, where φ is the dynamic modal formula we want to test for satisfiability (see Theorem 5). Since all the definitional clauses are range-restricted clauses, i.e. all variables of a clause occur in the negative part of the clause, all conclusions of hyperresolution and factoring inferences are ground clauses. Since in H_{sp} splitting is applied before the Deduce rules, all non-unit ground clauses are split before they are used as premises in inference steps. This means that the positive premises of any hyperresolution inference step are always ground clauses. These ground clauses have the form $Q_\psi(s)$ or $R_\alpha(t,u)$, where ψ is some dynamic modal formula, α is some relational formula and s,t,u are ground Skolem terms. $Q_\psi(s)$ and $R_\alpha(t,u)$ translate directly to the labelled formulae $s:\psi$ and $(t,u):\alpha$, where the s,t,u are now viewed as constants. We refer to $Q_\psi(s)$ and $R_\alpha(t,u)$ as the *formulae associated* with $Q_\psi(s)$ and $R_\alpha(t,u)$. Every hyperresolution inference step in H_{sp} involves one (or two) positive premises $C_1(,C_2)$ and a negative premise D from Figure 3. The positive premises C_1 and C_2 are ground, unit clauses of the form $Q_\psi(s)$ or $R_\alpha(t,u)$. Following from what we have just said, the conclusion is a positive clause again, and it is either a ground unit clause of the same form, or it is a positive, splittable clause of ground literals of that form. Now it does not take much to see how to write down the tableau rule which performs exactly the hyperresolution inference just described, possibly followed by splitting. In particular, take a definitional clause $C = \neg A_1[\,\vee \neg A_2] \vee D$, where A_1, A_2 denote atoms and D is the largest positive subclause of C. If C is a negative clause we let $D = \bot$. C contains at most two variables. Substitute these with s and t, i.e. apply the substitution $\sigma = \{x/s, y/t\}$ to C. Now write $C\sigma$ as the rule $F_1(,F_2)/G$, where F_1 and F_2 are the labelled formulae associated with $A_1\sigma$ and $A_2\sigma$. Similarly D becomes G, but if G is not a unit clause disjunction is replaced by $|$. For example, the definitional clause for $[\alpha]\psi$, $\neg Q_{[\alpha]\psi}(x)^+ \vee \neg R_\alpha(x,y)^+ \vee Q_\psi(y)$, is turned into the rule $s:[\alpha]\psi, (s,t):\alpha \,/\, t:\psi$. All rules in Figure 5 can be obtained in this way from Figure 3. Let Tab be the tableau calculus for $K_{(m)}(\wedge,\vee,{}^\smile,\upharpoonright)$ given by these rules.

LEMMA 9. *H_{sp} step-wise simulates Tab with respect to the structural transformation defined in Theorem 5.*

THEOREM 10. *There is a linear bisimulation between Tab and H_{sp} on the structural transformation for local satisfiability in $K_{(m)}(\wedge,\vee,{}^\smile,\upharpoonright)$.*

That H_{sp} p-simulates Tab is a direct consequence of Lemma 9. For simulation in the other direction we can show that there are no H_{sp}-refutations that contain any steps that have no counter-part in a tableau derivation. The only rule application that has no counter-part in Tab is the factoring rule, but since we assume that $\wedge$ is an idempotent operation, factoring is never applied. Assuming idempotency of $\wedge$ is closely related to assuming that clauses are sets of literals rather than multisets of literals. When viewing clauses as sets we know that when using hyperresolution on range-restricted clauses, factoring is optional for completeness. In the case of $K_{(m)}(\wedge,\vee,{}^\smile,\upharpoonright)$ the clauses we are using are range-restricted, and the issue of whether to make the idempotency assumption or add contraction rules to Tab is a minor technical matter. Implemented tableau theorem

$$(\bot)\ \frac{s:\psi, s:\neg\psi}{\bot}$$

$$(\wedge)_1\ \frac{s:\psi\wedge\phi}{s:\psi} \qquad (\wedge)_2\ \frac{s:\psi\wedge\phi}{s:\phi} \qquad (\neg\wedge)\ \frac{s:\neg(\psi\wedge\phi)}{s:\sim\psi \mid s:\sim\phi}$$

$$(\neg[\cdot])_1\ \frac{s:\neg[\alpha]\psi}{(s,t):\alpha} \qquad (\neg[\cdot])_2\ \frac{s:\neg[\alpha]\psi}{t:\sim\psi}$$

$$([\cdot])\ \frac{s:[\alpha]\psi, (s,t):\alpha}{t:\psi}$$

$$(\smile)\ \frac{(s,t):\alpha^{\smile}}{(t,s):\alpha} \qquad (\smile)_I\ \frac{(t,s):\alpha}{(s,t):\alpha^{\smile}}$$

$$(\wedge)^r_1\ \frac{(s,t):\alpha\wedge\beta}{(s,t):\alpha} \qquad (\wedge)^r_2\ \frac{(s,t):\alpha\wedge\beta}{(s,t):\beta} \qquad (\wedge)^r_I\ \frac{(s,t):\alpha, (s,t):\beta}{(s,t):\alpha\wedge\beta}$$

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

$$(\vee)^r_{I,1}\ \frac{(s,t):\alpha}{(s,t):\alpha\vee\beta} \qquad (\vee)^r_{I,2}\ \frac{(s,t):\beta}{(s,t):\alpha\vee\beta}$$

$$(\upharpoonleft)_1\ \frac{(s,t):\alpha\upharpoonleft\phi}{(s,t):\alpha} \qquad (\upharpoonleft)_2\ \frac{(s,t):\alpha\upharpoonleft\phi}{s:\phi} \qquad (\upharpoonleft)_I\ \frac{(s,t):\alpha}{(s,t):\alpha\upharpoonleft\phi \mid s:\sim\phi}$$

The side conditions of the $(\neg[\cdot])_j$ rules ($j \in \{1,2\}$) are that t is a constant uniquely associated with the premise $s:\neg[\alpha]\psi$. For the rules $(\smile)_I$, $(\wedge)^r_{I,j}$, $(\vee)^r_I$ and $(\upharpoonleft)_I$ the side conditions are that the relational formulae in the conclusions, occur as subformulae of the relational formula γ of a box formula $s:[\gamma]\psi$ on the current branch.

Figure 5. Tableau calculus for $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonleft)$.

provers use rewrite rules to replace obvious redundancies like duplication of the kind $\psi\wedge\psi$ even if no contraction rule is part of the calculus. Even when clauses are interpreted as multisets where duplication matters, factoring is optional for completeness for range-restricted clauses when using hyperresolution with splitting. However then identical branches might be created and potentially considerable work is repeated during the deduction process. Thus it does make sense for efficiency reasons to extend *Tab* with contraction rules. The following are the contraction rules corresponding to the factoring steps performed in H_{sp}.

$$(contr)\ \frac{s:\neg(\psi\wedge\psi)}{s:\sim\psi} \qquad (contr)^r\ \frac{(s,t):\alpha\vee\alpha}{(s,t):\alpha}$$

It is on conclusions with the definitional clauses for negated conjunctions of modal formulae and positive occurrences of disjunctions of relational formulae that factoring will be performed.

Soundness and refutational completeness of *Tab* (with or without contraction) is now a consequence of the soundness and refutational completeness of the H_{sp} and Theorem 5 (and Theorem 8(ii)).

The close link to resolution provides justification to enhance the tableau calculus with corresponding notions of redundancy [8]. We say a labelled formula F is *redundant* in a node, if the node contains labelled formulae $F_1, \ldots, F_n$ which are smaller than F and $\models_L (F_1 \wedge \ldots \wedge F_n) \rightarrow F$ (for $n \geq 0$). We can base the definition of an ordering on the subformula or subterm ordering, but a more general definition similar to admissible orderings in the resolution framework (cf. [3]) may be chosen. For example, all tautologies in L are redundant according to this definition. The application of a rule is *redundant* if its conclusions are redundant in the current node. For example, if a node includes $s : \neg p$ and $s : \neg(p \wedge q)$, then the $(\neg\wedge)$ rule need not be applied, and creating a new branch can be avoided. (The inference step is redundant because in the corresponding hyperresolution derivation the clauses $Q_{\neg p}(s)$ and $Q_{\neg(p\wedge q)}(s)$ are present and the conclusion $Q_{\neg p}(s) \vee Q_{\neg q}(s)$ with the definitional clause for $Q_{\neg(p\wedge q)}$ is subsumed by $Q_{\neg p}(s)$ and therefore redundant.) We can also use the link to resolution to introduce notions of *redundant tableau inference rules*. An inference rule is redundant with respect to a set X of labelled formulae and a calculus C if the definitional clause associated with the rule is redundant with respect to the union of all definitional clauses associated with the calculus (and X), and the clauses associated with the formulae in X. We can use this notion to define redundant inference rules in a calculus by letting $X = \emptyset$. Observe that this notion allows an inference rule which is not redundant in a calculus to be redundant with respect the calculus and some (derived) formulae.

THEOREM 11. *A formula φ is locally satisfiable in $K_{(m)}(\wedge, \vee, \smile, \upharpoonright)$ iff a tableau derivation containing a branch $\mathcal{B}$ can be constructed in Tab (modulo redundancy, and with or without contraction) such that $\mathcal{B}$ does not contain $\bot$ and each rule application is redundant.*

This result states soundness and completeness of the calculus Tab. Notice the formulation is stronger than those normally found in the literature. The result extends Theorem 8.1 in [8].

The calculus Tab is unusual for a tableau calculus in that it requires the rules for the relational operators to be applied in two directions. Since the calculus comprises both elimination rules and introduction rules, the calculus can also be viewed as a restricted form of natural deduction calculus. In general, uncontrolled use of introduction rules can jeopardise decidability, but the side conditions restrict the applications of the rules in such a way that no formulae are introduced that do not occur in the input problem. Notice that the side conditions follow from the clausal form of the translated problem. The side conditions imply that the calculus has the subformula property. Since the clausal form does not include a positive non-ground clause the calculus is cut-free. Consequently, any procedure based on this calculus is a decision procedure, and no loop detection mechanism is necessary to ensure termination.

COROLLARY 12. *Any (fair) procedure based on Tab is a decision procedure* *for local satisfiability in* $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$.

This result is also a direct consequence of Theorem 8(i). The following results are extensions of results in [8, 18].

Let N_∞ denote the *limit* of a path $(N =)N_0, N_1, \ldots$ in a resolution derivation starting with N. By definition, N_∞ is the set $\bigcup_{j\geq 0} \bigcap_{k\geq j} N_k$ of persistent clauses in the path.

LEMMA 13. *Let* φ *be any* $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$*-formula. Let* N *be the clausal form of the structural transformation of* φ*. Let* I *be the set of positive ground unit clauses in the limit* N_∞ *of a complete open branch in a* H_{sp}*-derivation starting with* N*. Then: (i)* I *is a (Herbrand) model of* N_∞ *and* N*, if* N_∞ *does not contain the empty clause. (ii) A* $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$*-model of* φ *can be read off from* I*.*

THEOREM 14. *(i) A finite modal model for any modal formula locally satisfiable in* $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$ *can be effectively constructed with any (fair) procedure based on* H_{sp}*. (ii)* $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$ *has the finite model property.*

Due to the exact correspondence between clauses and formulae in H_{sp}- and *Tab*-derivations we can conclude:

COROLLARY 15. *A finite modal model for any modal formula locally satisfiable in* $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$ *can be effectively constructed with any (fair) procedure based on Tab.*

The results for $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$ in this section hold for all dynamic modal logics defined over the operators $\{\wedge, \vee, ^{\smile}, \upharpoonright, \downharpoonright\}$.

5 Rasiowa-Sikorski Systems

The 'develop via first-order resolution' approach is not limited to the development of tableau procedures. In this section we use the approach to develop a calculus for validity testing. More specifically, we develop a Rasiowa-Sikorski proof calculus for the dynamic modal logic $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$ and show that it is a decision procedure and can be used as the basis for a generator of counter-models.

Rasiowa-Sikorski proof systems [25] are dual tableau systems [34]. Given a formula F, they aim to prove its validity, or, if F is not valid, they aim to construct a counter-model, i.e. a model for the complement of the formula. Starting with the given formula F, this is done by systematic case analysis until fundamental validities are found. Fundamental validities are normally obvious validities such as the law of excluded middle (i.e. $\neg F \vee F$). Rasiowa-Sikorski expansion rules have the same form, $X \,/\, X_1 | \ldots | X_n$, as tableau rules and are also applied top-down. The definition of a Rasiowa-Sikorski derivation, and its construction by application of rules, is the same as for a tableau derivation. There is a slight variation in notation and crucially the interpretation of the rules is different. More precisely, X, X_i denote sets of labelled formulae, as in the previous section, but sets of formulae are now interpreted as disjunctions of formulae, whereas branching is interpreted conjunctively.

$$(\neg\bot)\ \frac{s:\psi,\, s:\neg\psi}{\neg\bot}$$

$$(\neg\wedge)_1\ \frac{s:\neg(\psi\wedge\phi)}{s:{\sim}\psi} \qquad (\neg\wedge)_2\ \frac{s:\neg(\psi\wedge\phi)}{s:{\sim}\phi} \qquad (\wedge)\ \frac{s:\psi\wedge\phi}{s:\psi \mid s:\phi}$$

$$([\cdot])_1\ \frac{s:[\alpha]\psi}{(s,t):{\sim}\alpha} \qquad ([\cdot])_2\ \frac{s:[\alpha]\psi}{t:\psi}$$

$$(\neg[\cdot])\ \frac{s:\neg[{\sim}\alpha]\psi,\,(s,t):\alpha}{t:{\sim}\psi}$$

$$(\neg\smile)\ \frac{(s,t):\neg(\alpha^{\smile})}{(t,s):{\sim}\alpha} \qquad (\neg\smile)_I\ \frac{(t,s):\alpha}{(s,t):\neg(({\sim}\alpha)^{\smile})}$$

$$(\neg\wedge)^r_1\ \frac{(s,t):\neg(\alpha\wedge\beta)}{(s,t):{\sim}\alpha} \qquad (\neg\wedge)^r_2\ \frac{(s,t):\neg(\alpha\wedge\beta)}{(s,t):{\sim}\beta}$$

$$(\neg\wedge)^r_I\ \frac{(s,t):\alpha,\,(s,t):\beta}{(s,t):\neg({\sim}\alpha\wedge{\sim}\beta)}$$

$$(\neg\vee)^r\ \frac{(s,t):\neg(\alpha\vee\beta)}{(s,t):{\sim}\alpha \mid (s,t):{\sim}\beta}$$

$$(\neg\vee)^r_{I,1}\ \frac{(s,t):\alpha}{(s,t):\neg({\sim}\alpha\vee{\sim}\beta)} \qquad (\neg\vee)^r_{I,2}\ \frac{(s,t):\beta}{(s,t):\neg({\sim}\alpha\vee{\sim}\beta)}$$

$$(\neg\upharpoonright)_1\ \frac{(s,t):\neg(\alpha\upharpoonright\phi)}{(s,t):{\sim}\alpha} \qquad (\neg\upharpoonright)_2\ \frac{(s,t):\neg(\alpha\upharpoonright\phi)}{s:{\sim}\phi}$$

$$(\neg\upharpoonright)_I\ \frac{(s,t):\alpha}{(s,t):\neg(({\sim}\alpha)\upharpoonright\phi) \mid s:\phi}$$

The side conditions of the $([\cdot])_j$ rules ($j \in \{1,2\}$) are that t is a constant uniquely associated with the premise $s:[\alpha]\psi$. For the rules $(\smile)_I$, $(\wedge)^r_{I,j}$, $(\vee)^r_I$ and $(\upharpoonright)_I$ the side conditions are that the relational formulae in the conclusions, occur as subformulae of the relational formula γ of a box formula $s:[\gamma]\psi$ on the current branch.

Figure 6. Rasiowa-Sikorski calculus for $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonright)$.

A Rasiowa-Sikorski calculus for local validity in $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonright)$ is presented in Figure 6. Let the calculus be denoted by *RS*. Notice that the rules are dual to the rules of the tableau calculus *Tab* in Figure 5. To see this, inductively define a function g such that any labelled formula $s:\psi$ in a *Tab*-derivation, starting with $a:\varphi$, is mapped to $g(s):{\sim}\psi$ in the corresponding *RS*-derivation, starting with $a:{\sim}\varphi$ (where $g(a)=a$ and any term s which was introduced for $\neg[\alpha]\phi$ is mapped by g to the term introduced for $[\alpha]\phi$). In addition, any labelled relational formula $(s,t):\alpha$ is mapped to $(g(s),g(t)):{\sim}\alpha$. Then extend the definition to a mapping of *Tab*-inferences to *RS*-inferences. We can show:

LEMMA 16. *The calculi Tab and RS step-wise simulate each other in a dual manner.*

THEOREM 17. *There is a linear bisimulation between* Tab *and* RS *for local satisfiability/validity in* $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$.

A detailed analysis of the duality between tableau and Rasiowa-Sikorski calculi for Peirce logic can be found in [34].

Lemma 16 implies that all the properties of the tableau calculus transfer to the Rasiowa-Sikorski calculus. With the notion of redundancy dualised in the expected way, we can state:

COROLLARY 18. *A formula* φ *is locally valid in* $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$ *iff a* RS*-derivation containing a branch* $\mathcal{B}$ *can be constructed (modulo redundancy, and with or without the dual contraction rules) such that* $\mathcal{B}$ *does not contain* $\neg\bot$ *and each rule application is redundant.*

COROLLARY 19. *Any (fair) procedure based on* RS *is a decision procedure for local validity in* $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$.

COROLLARY 20. *A finite modal counter-model for any modal formula which is locally* $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$*-invalid can be effectively constructed with any (fair) procedure based on* RS.

It is also possible to obtain the rules of the RS-calculus via resolution; this time we use resolution in dual form. The dual form of resolution is not very well-known but a little reflection will convince the reader that it is an obvious alternative interpretation of resolution.

Dual resolution calculi operate exactly like resolution calculi with the difference that clauses are obtained by transformation into disjunction normal form and dual Skolemisation, i.e. Skolem terms are used to eliminate universal quantifiers. Also, the dual form of structural transformation is used. The dual clause form is a set of conjunctions of literals and the set is interpreted as a disjunction. The empty clause is interpreted as $\top$. The definition of dual ordered resolution with selection is exactly the same as $\mathsf{R}^{\mathrm{red}}_{\mathrm{sp}}$, except that the disjunction in clauses is viewed as a conjunction and branching in the splitting rule is interpreted conjunctively. For example the dual resolution rule for propositional logic derives $C \wedge D$ from $C \wedge A$ and $\neg A \wedge D$. In fact, all techniques and results of classical resolution carry over to dual resolution. As a consequence, by simply interpreting clauses dually, and also all transformation and derivation steps as well as all deletion steps, we can use any resolution prover as a dual resolution prover for validity testing of formulae.

Dual Structural Transformation into Clausal Form

Since we are now interested in showing the validity of a problem we need to base our reduction to first-order logic on the following *dual structural*

Subformula θ	Definitional dual clauses for $\mathrm{Def}^d(\theta)$
(shortcut)	$\neg Q_{\neg\psi}(x) \wedge \neg Q_{\psi}(x)$
$\psi \wedge \phi$	$\neg Q_{\psi\wedge\phi}(x) \wedge Q_{\psi}(x) \wedge Q_{\phi}(x)$
$\neg(\psi \wedge \phi)$	$\neg Q_{\neg(\psi\wedge\phi)}(x) \wedge Q_{\sim\psi}(x)$
	$\neg Q_{\neg(\psi\wedge\phi)}(x) \wedge Q_{\sim\phi}(x)$
$[\alpha]\psi$	$\neg Q_{[\alpha]\psi}(x) \wedge \neg R_{\alpha}(x, f_{[\alpha]\psi}(x))$
	$\neg Q_{[\alpha]\psi}(x) \wedge Q_{\psi}(f_{[\alpha]\psi}(x))$
$\neg[\alpha]\psi$	$\neg Q_{\neg[\alpha]\psi}(x) \wedge R_{\sim\alpha}(x,y) \wedge Q_{\sim\psi}(y)$
(shortcut)	$\neg R_{\neg\alpha}(x,y) \wedge \neg R_{\alpha}(x,y)$
	$R_{\neg\alpha}(x,y) \wedge R_{\alpha}(x,y)$
$\alpha \wedge \beta$	$\neg R_{\alpha\wedge\beta}(x,y) \wedge R_{\alpha}(x,y) \wedge R_{\beta}(x,y)$
	$R_{\alpha\wedge\beta}(x,y) \wedge \neg R_{\alpha}(x,y)$
	$R_{\alpha\wedge\beta}(x,y) \wedge \neg R_{\beta}(x,y)$
$\alpha \vee \beta$	$\neg R_{\alpha\vee\beta}(x,y) \wedge R_{\alpha}(x,y)$
	$\neg R_{\alpha\vee\beta}(x,y) \wedge R_{\beta}(x,y)$
	$R_{\alpha\vee\beta}(x,y) \wedge \neg R_{\alpha}(x,y) \wedge \neg R_{\beta}(x,y)$
$\alpha^{\smile}$	$\neg R_{\alpha^{\smile}}(x,y) \wedge R_{\alpha}(y,x)$
	$R_{\alpha^{\smile}}(x,y) \wedge \neg R_{\alpha}(y,x)$
$\alpha{\upharpoonright}\phi$	$\neg R_{\alpha\upharpoonright\phi}(x,y) \wedge R_{\alpha}(x,y) \wedge Q_{\phi}(x)$
	$R_{\alpha\upharpoonright\phi}(x,y) \wedge \neg R_{\alpha}(x,y)$
	$R_{\alpha\upharpoonright\phi}(x,y) \wedge Q_{\sim\phi}(x)$

Figure 7. Definitional dual clausal forms for $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$

translation of formulae in dynamic modal logic:

$$
\begin{aligned}
\mathrm{Def}^d(\psi) =^{\mathrm{def}}\ & \forall x\, (\pi(\psi, x) \rightarrow Q_{\psi}(x)) \wedge \forall x\, (\pi(\sim\psi, x) \rightarrow Q_{\sim\psi}(x)) \\
& \wedge \forall x\, (\neg Q_{\sim\psi}(x) \rightarrow Q_{\psi}(x)), \\
\mathrm{Def}^d(\alpha) =^{\mathrm{def}}\ & \forall x\, y\, (R_{\alpha}(x,y) \rightarrow \pi(\alpha, x, y)) \wedge \forall x\, y\, (\pi'(\alpha, x, y) \rightarrow R_{\alpha}(x,y)) \\
& \wedge \forall x\, y\, (R_{\sim\alpha}(x,y) \leftrightarrow \neg R_{\alpha}(x,y)).
\end{aligned}
$$

The mapping π is defined as in Section 3. We can prove:

THEOREM 21. *Let L be a dynamic modal logic defined over the operators $\{\wedge, \vee, ^{\smile}, \upharpoonright\}$, and let φ be any modal formula. Suppose N is the set of dual clauses obtained from $\varphi' =^{\mathrm{def}} \bigwedge \mathrm{Def}^d(\mathrm{Sf}(\varphi)) \rightarrow \forall x\, Q_{\varphi}(x)$ by transformation into disjunctive normal form, inner dual Skolemisation, and clausifying the Skolemised formula. Then: (i) Each clause in N is either a unit clause $Q(a)$, for some Skolem constant a, or it is an instance of a dual definitional clause given in Figure 7. (ii) φ is locally valid in L iff $\models \varphi'$ iff $\models N$.*

Let $\mathit{DH}_{\mathrm{sp}}$ denote the unordered dual hyperresolution calculus $\mathit{DH}^{\mathrm{red}}_{\mathrm{sp}}$ with splitting.

Subformula θ	Definitional dual clauses for $\mathrm{Def}^d(\theta)$
(shortcut)	$\neg Q_{\neg\psi}(x)^+ \wedge \neg Q_{\psi}(x)^+$
$\psi \wedge \phi$	$\neg Q_{\psi\wedge\phi}(x)^+ \wedge Q_{\psi}(x) \wedge Q_{\phi}(x)$
$\neg(\psi \wedge \phi)$	$\neg Q_{\neg(\psi\wedge\phi)}(x)^+ \wedge Q_{\sim\psi}(x)$ $\neg Q_{\neg(\psi\wedge\phi)}(x)^+ \wedge Q_{\sim\phi}(x)$
$[\alpha]\psi$	$\neg Q_{[\alpha]\psi}(x)^+ \wedge R_{\sim\alpha}(x, f_{[\alpha]\psi}(x))$ $\neg Q_{[\alpha]\psi}(x)^+ \wedge Q_{\psi}(f_{[\alpha]\psi}(x))$
$\neg[\alpha]\psi$	$\neg Q_{\neg[\alpha]\psi}(x)^+ \wedge \neg R_{\alpha}(x,y)^+ \wedge Q_{\sim\psi}(y)$
$\alpha \wedge \beta$	$R_{\neg(\alpha\wedge\beta)}(x,y) \wedge \neg R_{\sim\alpha}(x,y)^+ \wedge \neg R_{\sim\beta}(x,y)^+$ $\neg R_{\neg(\alpha\wedge\beta)}(x,y)^+ \wedge R_{\sim\alpha}(x,y)$ $\neg R_{\neg(\alpha\wedge\beta)}(x,y)^+ \wedge R_{\sim\beta}(x,y)$
$\alpha \vee \beta$	$R_{\neg(\alpha\vee\beta)}(x,y) \wedge \neg R_{\sim\alpha}(x,y)^+$ $R_{\neg(\alpha\vee\beta)}(x,y) \wedge \neg R_{\sim\beta}(x,y)^+$ $\neg R_{\neg(\alpha\vee\beta)}(x,y)^+ \wedge R_{\sim\alpha}(x,y) \wedge R_{\sim\beta}(x,y)$
$\alpha^{\smile}$	$\neg R_{\neg(\alpha^{\smile})}(x,y)^+ \wedge R_{\sim\alpha}(y,x)$ $R_{\neg(\alpha^{\smile})}(x,y) \wedge \neg R_{\sim\alpha}(y,x)^+$
$\alpha\upharpoonright\phi$	$R_{\neg(\alpha\upharpoonright\phi)}(x,y) \wedge \neg R_{\sim\alpha}(x,y)^+ \wedge Q_{\phi}(x)$ $\neg R_{\neg(\alpha\upharpoonright\phi)}(x,y)^+ \wedge R_{\sim\alpha}(x,y)$ $\neg R_{\neg(\alpha\upharpoonright\phi)}(x,y)^+ \wedge Q_{\sim\phi}(x)$

Figure 8. Definitional dual clausal forms for $K_{(m)}(\wedge, \vee, {}^{\smile}, \upharpoonright)$ in range-restricted form.

By duality, soundness and completeness for validity are a consequence of Corollary 3 (and also Theorem 2).

COROLLARY 22. $\mathit{DH}^{\mathrm{red}}_{\mathrm{sp}}$ *and* $\mathit{DH}_{\mathrm{sp}}$ *are sound and complete proof systems for sets of dual clauses.*

The same result is true for dual ordered hyperresolution and dual ordered resolution with selection calculi.

Looking at Figure 7 we note that the clauses are not all range-restricted. It is in particular the clause $\neg Q_{\neg[\alpha]\psi}(x) \wedge R_{\sim\alpha}(x,y) \wedge Q_{\sim\psi}(y)$ associated with subformulae of the form $\neg[\alpha]\psi$ that is not range-restricted. This is a minor issue, because using a standard shifting transformation it is possible to transform the clauses into range-restricted clauses. Shifting essentially switches the signs of all binary literals; more precisely, we can use shifting to replace $R_{\alpha}(s,t)$- and $\neg R_{\alpha}(s,t)$-literals by $\neg R_{\neg\alpha}(s,t)$- and $R_{\neg\alpha}(s,t)$-literals, respectively. This transforms the clauses into range-restricted clauses. The clauses obtained are given in Figure 8.

Alternatively, we take the input set N computed in accordance with Theorem 21. With ordered resolution restricted to inferences which involve at least one of the relational shortcut clauses

$$\neg R_{\neg\alpha}(x,y) \wedge \neg R_{\alpha}(x,y)^* \quad \text{and} \quad R_{\neg\alpha}(x,y) \wedge R_{\alpha}(x,y)^*$$

as a premise we obtain the clauses in Figure 8 as conclusions. We need to use an ordering under which the binary literals are larger than unary literals, and in the shortcut clauses the R_α-literals (indicated with a *) are maximal. As now no more inferences are possible on these, the relational shortcut clauses and clauses containing R_α-literals become redundant and can be deleted. Let N' denote the 'pre-saturated and purified' set of clauses obtained in this way from N. Since the obtained clause set is dual to the clause set obtained for satisfiability of the negated problem (compare Figures 7 and 8), we can now state:

THEOREM 23. *Let φ be any $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonright)$-formula and let N' be the set of clauses obtained by a pre-saturation and purification from the dual clausal form of the dual structural transformation. Then: (i) Any $\mathsf{DH}_{\mathrm{sp}}$-derivation from N' terminates. (ii) φ is locally valid in $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonright)$ iff the $\mathsf{DH}_{\mathrm{sp}}$-saturation of N' contains the empty clause $\top$.*

LEMMA 24. *$\mathsf{DH}_{\mathrm{sp}}$ step-wise simulates RS with respect to set N' obtained as defined in Theorem 23.*

THEOREM 25. *There is a linear bisimulation between RS and $\mathsf{DH}_{\mathrm{sp}}$ on the structural transformation for local validity in $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonright)$.*

Using duality and Theorems 8 and 14 we can now give alternative proofs of Corollaries 18–20, i.e. soundness and completeness of *RS*, decidability and finite counter-model generation for $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonright)$.

6 Modal Resolution Systems

Refutation calculi without a splitting or branching rule need an explicit representation of disjunction. This section shows that if we omit splitting from tableau simulating hyperresolution procedures we get (labelled) modal resolution calculi of the kind described in Areces et al [1].

We focus again on the dynamic modal logic $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonright)$. Figure 9 presents a calculus we can read off from the structural encoding in Figure 3 and hyperresolution without splitting. Observe how the closure rule ($\perp$) has become an instance of the standard, ground resolution rule. Because we view clauses as multisets it is necessary to add factoring rules. Note however positive factoring suffices for completeness. As clauses are always ground, factoring has the effect of eliminating duplicate literals from clauses. It is thus easy to see that it is not necessary to add factoring rules if clauses are viewed as sets. Denote the calculus defined in Figure 9 as *Res*.

LEMMA 26. *H (without splitting) step-wise simulates Res with respect to the structural transformation defined in Theorem 5.*

THEOREM 27. *There is a linear bisimulation between Res and H on the structural transformation for local satisfiability in $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonright)$.*

As before, the proofs exploit the correspondence between inference steps in the *Res*-derivation and the *H*-derivation on the structural transformation of the given modal formula.

$$(res)\ \frac{C \vee s : \psi,\, D \vee s : \neg\psi}{C \vee D}$$

$$(\wedge)_1\ \frac{C \vee s : \psi \wedge \phi}{C \vee s : \psi} \qquad (\wedge)_2\ \frac{C \vee s : \psi \wedge \phi}{C \vee s : \phi}$$

$$(\neg\wedge)\ \frac{C \vee s : \neg(\psi \wedge \phi)}{C \vee s :\sim \psi \vee s : \sim\phi}$$

$$(\neg[\cdot])_1\ \frac{C \vee s : \neg[\alpha]\psi}{C \vee (s,t) : \alpha} \qquad (\neg[\cdot])_2\ \frac{C \vee s : \neg[\alpha]\psi}{C \vee t : \sim\psi}$$

$$([\cdot])\ \frac{C \vee s : [\alpha]\psi,\, D \vee (s,t) : \alpha}{C \vee D \vee t : \psi}$$

$$(\smile)\ \frac{C \vee (s,t) : \alpha^{\smile}}{C \vee (t,s) : \alpha} \qquad (\smile)_I\ \frac{C \vee (t,s) : \alpha}{C \vee (s,t) : \alpha^{\smile}}$$

$$(\wedge)^r_1\ \frac{C \vee (s,t) : \alpha \wedge \beta}{C \vee (s,t) : \alpha} \qquad (\wedge)^r_2\ \frac{C \vee (s,t) : \alpha \wedge \beta}{C \vee (s,t) : \beta}$$

$$(\wedge)^r_I\ \frac{C \vee (s,t) : \alpha,\, D \vee (s,t) : \beta}{C \vee D \vee (s,t) : \alpha \wedge \beta}$$

$$(\vee)^r\ \frac{C \vee (s,t) : \alpha \vee \beta}{C \vee (s,t) : \alpha \vee (s,t) : \beta}$$

$$(\vee)^r_{I,1}\ \frac{C \vee (s,t) : \alpha}{C \vee (s,t) : \alpha \vee \beta} \qquad (\vee)^r_{I,2}\ \frac{C \vee (s,t) : \beta}{C \vee (s,t) : \alpha \vee \beta}$$

$$(\upharpoonleft)_1\ \frac{C \vee (s,t) : \alpha\upharpoonleft\phi}{C \vee (s,t) : \alpha} \qquad (\upharpoonleft)_2\ \frac{C \vee (s,t) : \alpha\upharpoonleft\phi}{C \vee s : \phi}$$

$$(\upharpoonleft)_I\ \frac{C \vee (s,t) : \alpha}{C \vee (s,t) : \alpha\upharpoonleft\phi \vee s : \sim\phi}$$

$$(fact)\ \frac{C \vee s : \psi \vee s : \psi}{C \vee s : \psi} \qquad (fact)^r\ \frac{C \vee (s,t) : \alpha \vee (s,t) : \alpha}{C \vee (s,t) : \alpha}$$

The side conditions of the $(\neg\Box_i)_j$ rules ($j \in \{1,2\}$) are that t is a constant uniquely associated with the premise $s : \neg\Box_i\psi$. For the rules $(\smile)_I$, $(\wedge)^r_{I,i}$, $(\vee)^r_I$ and $(\upharpoonleft)_I$ the side conditions are that the relational formulae in the conclusions, occur as subformulae of the relational formula γ of a box formula $s : [\gamma]\psi$ in the current clause set.

Figure 9. Modal resolution calculus for $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonleft)$.

COROLLARY 28. *Res is sound and refutationally complete for testing the local satisfiability of formulae in* $K_{(m)}(\wedge, \vee, \smile, \rceil)$.

This illustrates that the step from tableau-style systems to systems with an explicit resolution rule is not big. If we view the expansion rules as lazy translation to first-order logic then semantic tableau without splitting (i.e. modal resolution) can be viewed as hyperresolution with lazy translation to first-order logic.

What about decidability? We can prove Theorem 8(i) also holds for hyperresolution without splitting, and any refinement of hyperresolution without splitting.

THEOREM 29. *Let* φ *be any* $K_{(m)}(\wedge, \vee, \smile, \rceil)$*-formula and let* N *be the clausal form of the structural transformation. Then any H-,* H^{red}*-derivation from* N *terminates.*

This means that the calculus *Res* provides a decision procedure as well.

THEOREM 30. *Any (fair) procedure based on Res is a decision procedure for local satisfiability in* $K_{(m)}(\wedge, \vee, \smile, \rceil)$.

An advantage of the 'develop via first-order resolution' approach is that it is possible to transfer any refinements compatible with the simulating resolution procedure to the newly developed calculus. This may appear to be quite a strong claim, but let us now apply the 'develop via first-order resolution' approach by using ordered hyperresolution *OH*, instead of unordered hyperresolution *H*, and see which rules 'fall out'. Suppose the ordering used is an arbitrary ordering $\succ$ admissible in the sense of [3]. Recall the rules of ordered hyperresolution from Figure 2. Since the selection function overrides the ordering, the ordering does not change which literals are resolved upon in a negative premise. However since no literals can be selected in the positive premises, the ordering restricts inferences to literals strictly maximal with respect to the ordering. These restrictions transfer as follows to the modal resolution calculus. Suppose $\succ'$ is the ordering on labelled modal formulae that corresponds exactly to the ordering $\succ$ used in *OH*. Let *ORes* be the *ordered modal resolution calculus* based on $\succ'$ which is given by the rules of *Res* (cf. Figure 9) but which have side conditions which say that, for each rule except for the factoring rules, the explicitly given literals in every premise are strictly maximal with respect to $\succ'$. The side condition of the factoring rules is that the explicitly given literals are maximal with respect to $\succ'$. This means that any inferences that do not satisfy these side-conditions need not be performed (but can be, as is established in Corollary 28).

As before, it is possible to transfer any instances of redundancy elimination to the newly developed calculus. This includes tautology deletion, subsumption deletion and condensing. Similar as for tableau and Rasiowa-Sikorski systems we can define different forms of redundancy specifically for the modal resolution calculus. We refer to the extension of *ORes* with notions of redundancy as $\mathit{ORes}^{\mathrm{red}}$.

LEMMA 31. *OH ($\mathsf{OH}^{\mathrm{red}}$) based on an admissible ordering $\succ$ step-wise simulates ORes ($\mathsf{ORes}^{\mathrm{red}}$) based on the corresponding ordering $\succ'$ on labelled formulae with respect to the structural transformation defined in Theorem 5.*

THEOREM 32. *There is a linear bisimulation between $\mathsf{ORes}^{\mathrm{red}}$ and $\mathsf{OH}^{\mathrm{red}}$ on the structural transformation for local satisfiability in $K_{(m)}(\wedge, \vee, ^{\smile}, \rceil)$.*

THEOREM 33. *ORes and $\mathsf{ORes}^{\mathrm{red}}$ are sound and refutationally complete for testing the local satisfiability of formulae in $K_{(m)}(\wedge, \vee, ^{\smile}, \rceil)$.*

THEOREM 34. *Any (fair) procedure based on $\mathsf{ORes}^{(\mathrm{red})}$ is a decision procedure for local satisfiability in $K_{(m)}(\wedge, \vee, ^{\smile}, \rceil)$.*

Analogously, we can write down (unordered and ordered) modal dual resolution calculi for validity in $K_{(m)}(\wedge, \vee, ^{\smile}, \rceil)$ (with or without redundancy) and prove soundness, completeness and decidability by dual arguments. We leave it to the reader to work out the details.

7 Concluding Remarks

Of course, it is possible to come up with the calculi developed in this paper in an independent way and prove soundness, completeness and decidability based on traditional methods. By taking a slightly unusual approach we can however obtain new insights and results for (familiar and less familiar) logics and the different proof methods that could be used for theorem proving. The 'develop via first-order resolution' approach allows us to develop different calculi and directly compare them in a common framework. We have seen that small variations in the translation mapping and the resolution refinement used, result in notably different styles of calculi. For example, there is only a very small difference between the tableau calculus and the modal resolution calculus for $K_{(m)}(\wedge, \vee, ^{\smile}, \rceil)$ in the simulations (the omission of the splitting rule). This shows that Tab and Res are closely related. We have seen that by interpreting resolution dually we can even devise calculi for proving the validity of formulae. The approach allows us to transfer refinements and techniques such as redundancy elimination to the new calculi, and it enables the formulation of strong soundness and completeness results.

The approach is applicable to more expressive logics and straightforwardly yields sound and (refutationally) complete tableau, Rasiowa-Sikorski or resolution calculi for many, if not all, first-order definable (dynamic) modal logics. The approach applies also to other first-order definable logics and fragments of first-order logic. For example, linear bisimulation results between tableau systems and resolution have been obtained in [13] for decidable fragments of first-order logic which are closely related to the guarded fragment. It is clear that the 'develop via resolution' approach allows us to immediately pull out sound, refutationally complete and terminating tableau procedures for the generalisations obtained in the paper [13]. Similarly, tableau decision procedure can be defined for the solvable class $\mathcal{BU}$ introduced and studied in [12].

Soundness and completeness of the extracted calculi is in general automatic as long as the reduction to first-order logic and the refinement used have the properties identified in the Introduction so that bisimulation results can be proved. Apart from finding a suitable combination of a reduction and refinement, the main challenge is to develop ways to guarantee termination and prove decidability. In the case of $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$ there are existing decidability results for hyperresolution which we have extended.

It is possible to derive in a systematic way sound and complete calculi via non-standard translation methods, for instance, translations based on functional translation approaches (cf. [24, 33]) or the axiomatic translation approach [32]. In fact the results in [32] suggest that it is possible to obtain tableau calculi in a systematic and semi-automatic way from the Hilbert axiomatisations of traditional modal logics. The approach followed there also belongs to the 'develop via first-order resolution' approach. There is actually a lot of flexibility and potential in the approach which we have not explored due to lack of space.

There are good reasons why the approach discussed in this paper is combined with first-order resolution. Currently no other proof method exhibits the level of sophistication that the resolution framework has. It combines and integrates numerous principles and techniques that have been developed over many years in the area of automated reasoning. Previous work has shown that the available concepts of refinement and redundancy, in particular, mean that it is well suited for solving decidability problems of non-classical logics and fragments of first-order logic, cf. the surveys [10, 19, 33]. Furthermore, the framework is well suited for developing and studying model builders, cf. e.g. [5, 10, 11]. As we have seen in this paper refinements and redundancy are crucial ingredients of the 'develop via first-order resolution' approach.

One of the attractions of situating the approach within the framework of resolution is that it is easy to implement procedures based on the extracted calculi with existing first-order resolution theorem provers. Hyperresolution, or essentially equivalent refinements, are standardly implemented in all of the well-known first-order resolution theorem provers. Moreover, ordering restrictions can be flexibly defined in these and splitting is currently available in at least (M)SPASS [17, 36] and VAMPIRE [27]. With modest implementation effort it is therefore possible to use these provers as essentially modal tableau provers, modal Rasiowa-Sikorski provers, or modal resolution provers. All that is necessary, is to implement the structural transformations used and then choose the correct combination of flag settings so that the prover uses the simulating refinement. This provides a simple approach of implementing special-purpose procedures. In addition, it allows us to compare them also experimentally in a uniform framework. Examples of some empirical research performed in this framework can be found in [5, 18].

Further work consists of applying the approach to other logics, other styles of deduction methods as well as other forms of inference problems. We are presently investigating the simulation and development of natural deduction calculi for modal logics and first-order logic [28]. It would also be of interest if a resolution prover can be directly used as a special-purpose prover without any extra implementation effort. A student project to use MSPASS as a tableau prover for modal logics is currently under way.

Acknowledgements

The work has benefitted from discussions with Ullrich Hustadt and David Robinson. I also thank Yde Venema for useful comments.

BIBLIOGRAPHY

[1] C. Areces, M. de Rijke, and H. de Nivelle. Resolution in modal, description and hybrid logic. *J. Logic Computat.*, 11(5):717–736, 2001.

[2] F. Baader and U. Sattler. An overview of tableau algorithms for description logics. *Studia Logica*, 69:5–40, 2001.

[3] 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.

[4] 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.

[5] P. Baumgartner and R. A. Schmidt. Blocking and other enhancements for bottom-up model generation methods. In *Proc. IJCAR 2006*, LNAI. Springer, 2006. To appear.

[6] 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.

[7] H. de Nivelle. Splitting through new proposition symbols. In *Proc. LPAR 2001*, vol. 2250 of *LNAI*, pp. 172–185. Springer, 2001.

[8] H. De Nivelle, R. A. Schmidt, and U. Hustadt. Resolution-based methods for modal logics. *Logic J. IGPL*, 8(3):265–292, 2000.

[9] L. Fariñas del Cerro and O. Gasquet. A general framework for pattern-driven modal tableaux. *Logic J. IGPL*, 10(1):51–83, 2002.

[10] C. Fermüller, A. Leitsch, U. Hustadt, and T. Tammet. Resolution decision procedures. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, pp. 1791–1849. Elsevier, 2001.

[11] 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.

[12] 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.

[13] L. Georgieva, U. Hustadt, and R. A. Schmidt. Hyperresolution for guarded formulae. *J. Symbolic Computat.*, 36(1–2):163–192, 2003.

[14] 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.

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

[16] 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.

[17] 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.

[18] U. Hustadt and R. A. Schmidt. Using resolution for testing modal satisfiability and building models. *J. Automated Reasoning*, 28(2):205–232, 2002.
[19] U. Hustadt, R. A. Schmidt, and L. Georgieva. A survey of decidable first-order fragments and description logics. *J. Relational Meth. in Computer Sci.*, 1:251–276, 2004.
[20] F. Massacci. Single step tableaux for modal logics: Computational properties, complexity and methodology. *J. Automated Reasoning*, 24(3):319–364, 2000.
[21] F. Massacci. Decision procedures for expressive description logics with intersection, composition, converse of roles and role identity. In *Proc. IJCAI 2001*, pp. 193–198. Morgan Kaufmann, 2001.
[22] K. Nellas. Reasoning about sets and relations: A tableaux-based automated theorem prover for Peirce logic. Master's thesis, Univ. Manchester, UK, 2001.
[23] R. Nieuwenhuis and A. Rubio. Paramodulation-based theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, pp. 371–443. Elsevier, 2001.
[24] 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*, pp. 1403–1486. Elsevier, 2001.
[25] H. Rasiowa and R. Sikorski. *The Mathematics of Metamathematics*, vol. 41 of *Monografie matematyczne*. Polish Scientific Publ., Warsaw, 1963.
[26] A. Riazanov and A. Voronkov. Splitting without backtracking. In *Proc. IJCAI 2001*, pp. 611–617. Morgan Kaufmann, 2001.
[27] A. Riazanov and A. Voronkov. The design and implementation of VAMPIRE. *Artificial Intelligence Commun.*, 15(2-3):91–110, 2002.
[28] D. Robinson and R. A. Schmidt. Resolution and natural deduction. In preparation, 2006.
[29] J. A. Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12(1):23–41, 1965.
[30] 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.
[31] 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.
[32] R. A. Schmidt and U. Hustadt. The axiomatic translation principle for modal logic. *ACM Trans. Computational Logic*, 2005. To appear.
[33] R. A. Schmidt and U. Hustadt. First-order resolution methods for modal logics. In *Volume in Memoriam of Harald Ganzinger*, LNCS. Springer, 2006. To appear.
[34] 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.
[35] S. Schulz. E: A brainiac theorem prover. *J. AI Comm.*, 15(2–3):111–126, 2002.
[36] 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.

Renate A. Schmidt

School of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, UK.

Renate.Schmidt@manchester.ac.uk