Annals of Mathematics and Artificial Intelligence manuscript No.
(will be inserted by the editor)

A New Methodology for Developing Deduction Methods

Renate A. Schmidt

Received: date / Accepted: date

Abstract This paper explores the use of resolution as a meta-framework for developing various, different deduction calculi. In this work the focus is on developing deduction calculi for modal dynamic 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 can be developed and studied by using standard principles and methods of first-order theorem proving. The approach is based on the translation of reasoning problems in modal logic to first-order clausal form and using a suitable refinement of resolution to construct and mimic derivations of the desired proof method. The inference rules of the calculus can then be read off from the clausal form. We show how this approach can be used to generate new proof calculi and prove soundness, completeness and decidability results. This slightly unusual approach allows us to gain new insights and results for familiar and less familiar logics, for different proof methods, and compare them not only theoretically but also empirically in a uniform framework.

Keywords deduction calculus synthesis · resolution · tableaux · dual resolution · dual tableaux · modal resolution · decidability · modal logic

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 [10, 19, 25, 26, 28, 46, 48]. 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 of first-order resolution theorem proving. In [26] we have shown how derivations and search in standard tableau algorithms of the description logic $\mathscr{ALC}$ 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 [25] how to mimic and strengthen standard tableau algorithms for $\mathscr{ALC}$ with respect to non-empty TBoxes. This corresponds to

Renate A. Schmidt
School of Computer Science, The University of Manchester, Oxford Road, Manchester M13 9PL, UK
E-mail: Renate.Schmidt@manchester.ac.uk

local satisfiability in multi-modal logic $K_{(m)}$ with respect to a background theory of modal logic formulae (that is, a set of non-logical axioms or global assumptions). In [28] details can be found of how to simulate derivations in the prefixed single-step tableau calculi of Massacci [32]. These simulation results show that it is possible to use first-order resolution in a way that it closely 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 [10] 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 (constructors) can be relational formulae, which are interpreted as actions or programs in *PDL*, and are closely related to description logics [47]. $K_{(m)}(\wedge,\vee,\smile)$ 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 $\mathscr{ALC}$ in which conjunction, disjunction and converse of roles are allowed. In [10] we have shown how a tableau calculus can essentially be ‘read off’ from the clausal form of the translation of formulae in $K_{(m)}(\wedge,\vee,\smile)$. In [46, 48] we use resolution methods to develop a new translation mapping (called the axiomatic translation) of traditional style modal logics. This is then used to derive tableau inference rules, resulting in tableau calculi based on propagation rules rather than structural rules.

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 [30, 38]. In addition, we consider the development of modal resolution systems [1, 13, 15] 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 three aspects:

1. An effective, sound and complete translation to first-order logic that retains enough information about the input formula of the source logic so that the inference rules can be read off from the clausal form.
2. A refinement of first-order resolution which performs inferences exactly like the kind of system we want to develop.
3. If needed, partial pre-saturation and purification of the characteristic clauses.

The form and property of the calculus one obtains depends very much on all three aspects and small modifications result in different variations of calculi and also different styles 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 in the area, and many modal and description logic theorem provers are based on ground semantic calculi.

In order for the resolution inference steps to be translatable back into inference steps on modal formulae we need to use a translation to first-order logic that allows us to relate clauses back to modal formulae. This can be achieved by translation mappings combined with structural transformation. Structural transformation is a standard techniques in automated reasoning which introduces new predicate symbols and definitions. The cited previous work shows that ground semantic tableau calculi can be simulated by a structural transformation 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 transformation, allows us to interpret the negative, non-ground premise as an inference rule $\mathscr{I}$ of the ground calculus. The positive, ground premises of a hyperresolution inference step represent then the premises of the rule $\mathscr{I}$, and the conclusions represent the conclusions of the rule. Combined with splitting, hyperresolution allows us to simulate and develop ground semantic tableau calculi. We discuss how dual hyperresolution with splitting allows us to simulate and develop ground semantic Rasiowa-Sikorski calculi. Furthermore, we see how hyperresolution without a splitting rule produces modal resolution calculi.

To illustrate the approach we focus on the development of calculi for the dynamic modal logic $K_{(m)}(\wedge,\vee,\smile,\upharpoonright)$. In $K_{(m)}(\wedge,\vee,\smile,\upharpoonright)$ the additional relational operators are conjunction, disjunction, converse and domain restriction. This logic was chosen because when this work was undertaken for [43], which is the short version of the present paper, it has not been considered before in the literature and no deduction calculi had been described for it. $K_{(m)}(\wedge,\vee,\smile,\upharpoonright)$ is subsumed by Peirce logic, the logical version of Peirce algebras [6], for which tableau calculi are defined in [35, 49]. However, the relational disjunction operator and the domain restriction operator of $K_{(m)}(\wedge,\vee,\smile,\upharpoonright)$ are not explicit operators in Peirce logic. This means there are no tableau rules for these operators in the existing tableau calculi for Peirce logic. Although sound rules can be easily defined for these operators, 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, $K_{(m)}(\wedge,\vee,\smile,\upharpoonright)$ is decidable and Peirce logic is not decidable. Since this work was undertaken, tableau decision procedures have been developed in [50, 51] for description logics which subsume $K_{(m)}(\wedge,\vee,\smile,\upharpoonright)$. Since the logic $K_{(m)}(\wedge,\vee,\smile,\upharpoonright)$ is simple and has not been studied explicitly before, it is nevertheless still a good candidate for illustrating and exploring the possibilities of synthesising deduction calculi as an application of resolution.

This paper is of theoretical and practical interest, not only to researchers interested in modal and description logics, but also anybody interested in proof methods, decision procedures, the relationship between proof methods and developing such methods, and implementing theorem provers. Novel contributions are the following.

- The simulation of Rasiowa-Sikorski and modal resolution methods by first-order resolution.
- The interpretation of tableau, Rasiowa-Sikorski and modal resolution methods as specialisations (reductions) of first-order resolution.
- The use of a non-standard approach to developing proof methods and implemented systems.
- The application of resolution not merely as an automated reasoning procedure, but as a meta-framework within which it is possible to develop tableau calculi and other styles of calculi, decision procedures and even implemented provers.

- A uniform methodology for comparing different calculi, both with respect to theoretical and practical aspects.
- The transfer of refinements of resolution, such as orderings, to other calculi resulting in more efficient inference systems. In earlier work [25] we showed that a technique called absorption [23] used to limit the search space for globally satisfiable formulae in description logic tableau systems can be strengthened with hyperresolution.
- We show that the notion of redundancy, which is crucial for resolution but virtually absent in other calculi, carries over to other calculi, resulting in stronger results and improved inference systems. Similar to resolution, we define redundant formulae and redundant applications of inference steps in other calculi. In addition, we introduce a new notion of *redundant rules of inference*.
- The characterisation of the relationship between different deduction calculi and procedures in terms of the notions of simulation and reduction. If two calculi or procedures are correlated by a step-wise simulation and reduction relationship this defines an exact correspondence and makes clear that derivations in these two calculi are essentially the same, i.e. their behaviour is the same. As a consequence we may apply all available knowledge of one calculus (procedure) to infer properties of the corresponding other calculus (procedure). Of greater significance is that the method allows us to develop new and improved calculi.
- For the logics under consideration the methodology allows us not only to read off the inference rules of a calculus, it also allows us to determine a priori the maximum number of times a particular inference rule of the new calculus would need to be applied in a derivation for a given input formula. By inspecting the clausal set of the translation it is possible to say which inference rules are not needed for a given formula, and it is possible to tell the maximum number of applications of any inference rule.
- It is possible to tailor different styles of deduction approaches for different logics.

The rest of the paper is structured as follows. In the next section we recall standard definitions of resolution-based theorem proving. Section 3 discusses dynamic modal logics and defines $K_{(m)}(\wedge,\vee,{}^{\smile},\upharpoonright)$. Section 4 introduces and analyses the simulation and reduction mappings which are used to describe the relationship between different deduction approaches. It also defines the conversion to clausal form that we use. Section 5 defines the notions of simulation and reduction which allow us to transfer soundness, completeness and termination of hyperresolution to the obtained calculi. In Sections 6–8 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 some consequences of the method and mentions future work.

Throughout the paper we use the notation and terminology of our previous papers, see, for instance, the surveys [45, 47]. The paper is an extended and improved version of [43].

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. We assume that disjunction is a commutative and associative operator. This means that clauses are regarded as multisets of literals rather than sets of literals.

Theorem 1 *There is a linear reduction* Cls *of any first-order formula to clause logic such that φ is valid in first-order logic, that is $\models \varphi$, 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, that is, 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* [42], 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 [53], E [52] and VAMPIRE [40] (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 [3, 36].

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, tautology 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

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 and both are positive.

Resolvents and factors are computed with these rules.

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

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

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

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

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

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 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 [9, 39] implemented in the theorem prover VAMPIRE [40] or the generalisation called separation in [44].

The restriction that only positive clauses are split is not essential for the soundness and (refutational) completeness results below. The restriction is however important for our application.

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

Theorem 2 (Bachmair et al [3, 4]) *$R^{\text{red}}_{\text{sp}}$, R^{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 $OH^{\text{red}}_{\text{sp}}$ 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, OH^{red} denotes the calculus $OH^{\text{red}}_{\text{sp}}$ but without the splitting rule, OH denotes the calculus just consisting of Deduce rules, and OH_{sp} is OH with splitting. For completeness an ordering refinement is optional. We use the notation $H^{\text{red}}_{\text{sp}}$, H^{red}, H and H_{sp} for the unordered versions.

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

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

Fig. 2 The Deduce rules of ordered hyperresolution.

Corollary 1 *$OH^{\mathrm{red}}_{\mathrm{sp}}$, OH^{red}, OH_{sp}, OH, $H^{\mathrm{red}}_{\mathrm{sp}}$, H^{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 [47].

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

$$\bot,\ \neg\phi,\ \phi \wedge \psi, \text{ and } [\alpha]\phi$$

are dynamic modal formulae, and

$$\alpha \wedge \beta,\ \alpha \vee \beta,\ \alpha^{\smile} \text{ and } \alpha\upharpoonright\phi$$

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 $\mathscr{M} = (W, R, v)$, 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 $R(\alpha)$):

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

In addition, the following conditions are satisfied:

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

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

If $\mathscr{M},x \models \varphi$ holds then φ is *(locally) true* at x in $\mathscr{M}$ and $\mathscr{M}$ *(locally) satisfies* φ. A modal formula φ is *(locally) satisfiable* iff there exists a model $\mathscr{M}$ and a world x in $\mathscr{M}$ such that $\mathscr{M},x \models \varphi$. 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 [26] and also of the decidability of the two-variable fragment of first-order logic.

Theorem 3 *Let L be a dynamic modal logic with any subset of* $\{\wedge, \vee, ^\smile, \upharpoonleft, \downharpoonleft\}$ *as relational operators. The local (and global) satisfiability problem in L is decidable.*

It follows from [26, 33] that the result remains true when we allow negation as a relational operator [26], or relational composition and identity [33]. However, adding both relational negation and relational composition leads to undecidability. More precisely, any dynamic modal logic with relational conjunction, relational negation and (negative occurrences of) composition is undecidable [45, 47].

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 [46, 48]; other structural translations have been used in [10, 25, 26, 28], for instance.

Throughout the paper, we assume that all occurrences of double negation are eliminated from modal formulae. For any formula F, let $\sim F$ denote G if $F = \neg G$, and $\neg F$ otherwise. Thus, $\sim F$ is the complement of F.

Let Def be the transformation of dynamic modal formulae and relational formulae which is defined as follows.

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

$\mathrm{Def}(\psi)$ 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 and π' are defined in Figure 3. 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.

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

In $\pi([\alpha]\psi,x)$ and $\pi(\neg[\alpha]\psi,x)$, z denotes any variable distinct from x.

Fig. 3 Definition of the translation mappings π and π'.

Theorem 4 *Let L be a dynamic modal logic defined over the operators in $\{\wedge,\vee,{}^{\smile},\upharpoonright\}$, 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. Then:*

1. *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 4.*
2. *φ is locally satisfiable in L iff φ' is first-order satisfiable iff N is first-order satisfiable.*
3. *N is computable in linear time.*

The markings by $^{+}$ of some of the literals in Figure 4 can be ignored for now; they are explained in Section 6.

In line with [46, 48] the first clause in Figure 4 is called a *negative shortcut clause*. Intuitively, shortcut clauses or shortcut formulae link a formula to its negation. The remaining clauses are said to be *definitional clauses associated with the modal subformula θ* in the index of the first literal.

For efficiency reasons it is sensible to take the polarities of all the occurrences of a subformula in the input problem into account in the specification of Def. We do not do this here because the methodology considers all possible clausal forms that can be obtained for the logic under consideration, so that all possible inference rules for the logic can be extracted from the clausal forms.

4 Simulation and Reduction

In this section let L_1 and L_2 denote two logics (not necessarily modal logics). Suppose Π is a sound and complete translation of (sets of) formulae in L_1 to (sets of) formulae in L_2, that is, $N \models_1 \varphi$ iff $\Pi(N) \models_2 \Pi(\varphi)$ for any set $N \cup \{\varphi\}$ of formulae in L_1. In addition we assume

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

Fig. 4 Definitional clausal forms for $K_{(m)}(\wedge, \vee, {}^{\smile}, \upharpoonright)$.

that Π is computable in linear or polynomial time. Let C_1 be a calculus, or proof procedure, for L_1, and let C_2 be a calculus, or proof procedure, for L_2.

In the previous version of this paper [43] the transfer of soundness and completeness was based on the notion of p-simulation [8]. Formally, C_2 *p-simulates* (*proofs in*) C_1 (*with respect to* Π) iff the following condition holds: There is a function g computable in polynomial time which maps any proof (or refutation) in C_1 to a proof (or refutation) in C_2. If g maps a proof to a refutation, or a refutation to a proof, we say that C_2 *dually p-simulates* C_1.

In the next theorem and subsequently, when we use the notions 'complete' or 'completeness' in conjunction with a refutational calculus we mean 'refutationally complete' or 'refutational completeness'.

Theorem 5 *Suppose C_2 (dually) p-simulates C_1 (with respect to Π). Then*

1. *If C_1 is complete then C_2 is complete.*
2. *If C_2 is sound then C_1 is sound.*

Proof Suppose C_1 and C_2 are both calculi (procedures) for proving validity. Assume $N \models_1 \varphi$. Then by completeness of C_1 we have that $N \vdash_1 \varphi$. This implies $\Pi(N) \vdash_2 \Pi(\varphi)$, since C_2 p-simulates C_1 with respect to Π, and Π is sound and complete by assumption. This proves completeness. For soundness, assume $N \vdash_1 \varphi$. This implies $\Pi(N) \vdash_2 \Pi(\varphi)$ as above, and then by soundness of C_2 we get $\Pi(N) \models_2 \Pi(\varphi)$, which implies $N \models_1 \varphi$. The cases where one of C_1 and C_2, or both, are refutation calculi (procedures) are simple corollaries. □

Corollary 2 *If C_1 and C_2 p-simulate each other (with respect to Π), then C_1 is sound and (refutationally) complete iff so is C_2.*

The notion of p-simulation is sufficient for obtaining soundness and (refutational) completeness for the derived calculi from the soundness and refutational completeness of hyperresolution. As said, this was done in [43]. It is however more illuminating to use the stronger notions of *step-wise simulation* [11] and *reduction*. These notions can be used to correlate any derivations in different calculi, not just successful derivations that are proofs. We need these notions because our aim is to simulate the search performed by different procedures, especially decision procedures. This means that all inference steps need to be simulated; not just those leading to proofs.

By definition, C_2 *step-wise simulates* C_1 (*with respect to* Π) iff there is an n and a function g mapping every inference step in C_1 to a sequence of at most n inference steps in C_2 which derive the corresponding conclusion. More precisely, if N_i and N_{i+1} are consecutive sets in any C_1-derivation then $\Pi(N_{i+1})$ can be obtained by at most n inference steps in C_2 from $\Pi(N_i)$. If rather than '$N \models_1 \varphi$ iff $\Pi(N) \models_2 \Pi(\varphi)$' we have that '$N \models_1 \varphi$ iff $\Pi(N) \wedge \neg\Pi(\varphi)$ is unsatisfiable in L_2', then we say C_2 *step-wise simulates* C_1 (*with respect to* Π) *in a dual manner* (or C_2 *dually step-wise simulates* C_1).

The definition says that, if C_2 step-wise simulates C_1 every inference step in C_1 can be mimicked by a bounded number of inference steps in C_2 which produce the corresponding conclusion. Notice that a simulation in calculus C_2 may use more inference steps.

Theorem 6 *Suppose C_2 (dually) step-wise simulates C_1 (with respect to Π). Then*

1. *If C_1 is complete then C_2 is complete.*
2. *If C_2 is sound then C_1 is sound.*

The proof is the same as for Theorem 5. Thus, like p-simulation mappings, step-wise simulation mappings preserve soundness and (refutational) completeness. The more important property for the present paper is the transferal of soundness to the simulated calculus (the second property), because hyperresolution takes the role of C_2.

The notion of a reduction, or specialisation, is a mapping in the other direction. Formally, we say a calculus or procedure C_1 is a *reduct* (or *specialisation*) *of* C_2 (*with respect to* Π) iff there is an n such that the inference steps in any C_2-derivation can be uniquely and exhaustively grouped into macro inference steps of maximal length n that correspond to inference steps in C_1. More precisely, any complete or closed C_2-derivation can be partitioned into (possibly interleaving) sequences of sets with length $\leq n$ and there is a function h which maps each sequence $N_{i_1}, \ldots, N_{i_m}$ (where $m \leq n$ and $i_j < i_k$, if $j < k$) to one inference step in C_1 with corresponding premises and conclusions. n is the finite bound on the size of a macro inference step. If rather than '$N \models_1 \varphi$ iff $\Pi(N) \models_2 \Pi(\varphi)$' we have that '$N \models_1 \varphi$ iff $\Pi(N) \wedge \neg\Pi(\varphi)$ is unsatisfiable in L_2', then we say C_2 is a *dual reduct* of C_1 (*with respect to* Π).

Intuitively, an inference step in the reduct C_1 captures a combination of smaller inference steps in C_2. That is, we can think of the reduct using macro inference steps (or hyperinference steps). Inference steps in C_2 are finer, while inference steps in C_1 are coarser. The reduct can therefore be viewed as a specialisation of the other calculus.

Theorem 7 *Suppose C_1 is a (dual) reduct of C_2 (with respect to Π). Then*

1. *If C_2 is complete then C_1 is complete.*
2. *If C_1 is sound then C_2 is sound.*

Proof Similar to the proof of Theorem 5. □

The preservation of completeness by the reduction mapping is important for our methodology.

If C_2 (dually) step-wise simulates C_1 (with respect to Π) and the inverse of the step-wise simulation function is a (dual) reduction (with respect to Π) then we say that C_1 and C_2 (dually) *correspond* to each other (with respect to Π).

Corollary 3 *Suppose C_1 and C_2 (dually) correspond to each other (with respect to Π). If C_2 is sound and complete, then C_1 is sound and complete.*

This result is also true for calculi that p-simulate each other. We can prove:

Lemma 1 *1. If C_2 (dually) step-wise simulates C_1 then C_2 (dually) p-simulates C_1.*
2. If C_1 is a (dual) reduct of C_2 then C_1 (dually) p-simulates C_2.

The converse is not true in either case, because the p-simulation functions map proofs to proofs whereas the other two notions map derivations to derivations (i.e. not just derivations which are proofs). Theorems 6 and 7 are easy consequences of this lemma and Theorem 5.

Lemma 2 *If C_1 is a (dual) reduct of C_2 (with respect to Π) and n in the definition of the reduction mapping is 1 then C_1 (dually) step-wise simulates C_2 (with respect to Π).*

We use the next theorem to transfer decidability results to the derived calculi.

Theorem 8 *If C_1 and C_2 (dually) correspond to each other and C_2 is a decision procedure, then C_1 is a decision procedure.*

5 Simulating Tableau Systems

Numerous tableau methods have been developed, studied and also implemented for traditional modal logics, see, for instance, [7, 20, 32]. Tableau methods for some dynamic modal logics or logics equivalent to dynamic modal logics can be found in the description logic literature, see, for instance, [2, 33] and also here [10, 35, 49].

In this section, let us first look at how resolution can step-wise simulate ground semantic tableau for local satisfiability in the basic multi-modal logic $K_{(m)}$ and how tableau can be viewed as reductions of hyperresolution.

A ground semantic tableau calculus for $K_{(m)}$ is given by the rules in Figure 5. There are slight differences to similar calculi found in the literature (see, for instance [12, 20]). The rule (cl) is the closure rule and ($contr$) is the contraction rule. The remaining inference rules are called tableau *expansion rules*. Often standard definitions do not include contraction rules. Because we do not assume that conjunction is an idempotent operator, our calculus does include a contraction rule. Often only one rule for conjunctions and negated box formulae is included in calculi, but for reasons which become obvious, once we look at the simulation by resolution, we choose the given definitions. Both are insignificant variations. Observe that instead of introducing constants in the $(\neg\Box_i)_j$ rules, we can equally well use Skolem terms.

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 inference rules. Inference 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 inference 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. Then 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. We also stipulate that no inference step is performed that yields a formula already on the current branch.

$$\begin{array}{lll}
(cl)\ \dfrac{s:\psi,\ s:\neg\psi}{\bot} & (\bot)\ \dfrac{s:\bot}{\bot} & \\[2ex]
(\wedge)_1\ \dfrac{s:\psi\wedge\phi}{s:\psi} & (\wedge)_2\ \dfrac{s:\psi\wedge\phi}{s:\phi} & (\neg\wedge)\ \dfrac{s:\neg(\psi\wedge\phi)}{s:\sim\psi \mid s:\sim\phi} \\[2ex]
(\neg\Box_i)_1\ \dfrac{s:\neg\Box_i\psi}{(s,t):R_i} & (\neg\Box_i)_2\ \dfrac{s:\neg\Box_i\psi}{t:\sim\psi} & (\Box_i)\ \dfrac{s:\Box_i\psi,\ (s,t):R_i}{t:\psi} \\[2ex]
(contr)\ \dfrac{s:\neg(\psi\wedge\psi)}{s:\sim\psi} & &
\end{array}$$

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 $s:\neg\Box_i\psi$.

Fig. 5 Tableau calculus for $K_{(m)}$.

Recall that H_{sp} denotes the unordered hyperresolution calculus with splitting. From now on we assume that factoring, splitting, and the hyperresolution rule are applied in this order. This is an important assumption for the simulations and reductions. We also assume that no inference step is performed that yields a conclusion already present in the current set of clauses.

Now consider the clausal form of the structural translation of φ as defined in Theorem 4, that is, Figure 4. 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 the basic modal logic $K_{(m)}$ are in the top-half of the table in Figure 4. Notice that for each of these definitional clauses there is one corresponding tableau expansion or closure rule in Figure 5, and vice versa. For the contraction rule there is however no corresponding definitional clause.

The connection between the tableau calculus and H_{sp} is the following. Every application of a tableau rule can be step-wise simulated by one or two H_{sp}-inference steps. If φ is the input formula, the derivation in H_{sp} starts with an inference with the (only positive) clause $Q_\varphi(a)$ (see Theorem 4). This corresponds to the root of the tableau derivation given by $\{a:\varphi\}$. In the tableau derivation we pick a formula $s{:}\psi$ from a branch and attempt to apply one of the inference rules to it. If the branch contains a formula $s:\sim\psi$ the closure rule is applicable and yields $\bot$. In H_{sp} this derivation corresponds to resolving the clauses $Q_\psi(s)$ and $Q_{\sim\psi}(s)$ with the shortcut clause $\neg Q_{\neg\psi}(x)^+ \vee \neg Q_\psi(x)^+$ using one hyperresolution inference step. The resolvent is the empty clause $\bot$. In a similar fashion we can show that possible applications of each of the other tableau expansion rules, except for the $(\neg\wedge)$ rule, can be simulated by one hyperresolution inference step involving the appropriate definitional clause as negative premise. In the case of negated conjunction we need to follow a hyperresolution inference step with the corresponding definitional clause by a splitting step. An application of the contraction rule is simulated by one factoring step in H_{sp} on the conclusion of an inference step with the definitional clauses for negated conjunctions of modal formulae.

Lemma 3 *H_{sp} on the structural transformation defined in Theorem 4 step-wise simulates ground labelled tableau for local satisfiability in $K_{(m)}$ of Figure 5.*

Proof By an inductive proof that shows we can map any tableau derivation to a corresponding H_{sp}-derivation for the translated problem using an argument as sketched above. □

To prove that the inverse of the step-wise simulation mapping is a reduction mapping we need to show that all steps possible in H_{sp} belong to a group of steps involved in simulating tableau steps. We can indeed show that there are no H_{sp}-derivations that contain any steps that have no counter-part in the corresponding tableau derivation. Therefore:

Lemma 4 *The ground labelled tableau for local satisfiability in $K_{(m)}$ of Figure 5 is a reduct of H_{sp} on the structural transformation defined in Theorem 4.*

We can conclude that:

Theorem 9 *There is a linear correspondence between the tableau calculus in Figure 5 and H_{sp} on the structural transformation for local satisfiability in $K_{(m)}$.*

This strengthens a result in [10]. Related (but different) simulation results have been shown in [28] (see also [45]) 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 [46, 48] the results can be strengthened to prefixed single-step tableau calculi of other traditional modal logics.

6 Developing Tableau Systems for $K_{(m)}(\wedge,\vee,{}^{\smile},\upharpoonright)$

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

We can prove the following:

Theorem 10 *Let φ be an arbitrary $K_{(m)}(\wedge,\vee,{}^{\smile},\upharpoonright)$-formula and let $N = \mathrm{Cls}(\exists x\, Q_\varphi(x) \wedge \bigwedge \mathrm{Def}(\mathrm{Sf}(\varphi)))$ be the clausal form of the structural transformation defined in Theorem 4. Then:*

1. *Any H_{sp}-derivation from N terminates.*
2. *φ is locally unsatisfiable in $K_{(m)}(\wedge,\vee,{}^{\smile},\upharpoonright)$ iff the H_{sp}-saturation of N contains the empty clause.*

Proof Part 1 can be shown using an argument similar to the proof of Theorem 6.6 in [28] and Theorem 7.7 in [10]. Part 2 is a consequence of Theorem 4 and Corollary 1. □

The definitional clauses in the input set N have the form as specified in Figure 4. The literals selected by the selection function of the calculus H_{sp} are marked with $^+$. The only other clause in an 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. Note that $Q_\varphi(a)$ is the only positive clause in N.

A crucial property is that the clauses in N are all *range-restricted* clauses, that is, all variables of a clause occur in the negative part of the clause.[1] Hyperresolution on range-restricted clauses has the property that all conclusions of hyperresolution and factoring inferences are ground clauses. Positive range-restricted clauses are always ground. This implies that factoring and splitting in H_{sp} are applied only to positive *ground non-unit* clauses. Since factoring and splitting are applied before the hyperresolution rule, all non-unit ground clauses are either factored and then split or just split, before they are used as premises in hyperresolution inference steps. This means that the positive premises of any hyperresolution inference step in H_{sp} are always *ground unit* clauses.

[1] Range-restricted clauses and the range restriction operator are not related in any way.

For the class of clauses we are working with, these ground unit 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 $s:\psi$ and $(t,u):\alpha$ 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 4. 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 clause of ground literals of that form which can be factored or split. We also observe that, since $Q_\varphi(a)$ is the only positive clause in N, the first inference step in any H_{sp}-derivation is performed with this clause.

Now it does not take much to see how we can write down the tableau rules which perform exactly the H_{sp}-inference steps just described. 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, that is, 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 then disjunction is replaced by $|$. For example, the definitional clause for $[\alpha]\psi$,

$$\neg Q_{[\alpha]\psi}(x)^+ \vee \neg R_\alpha(x,y)^+ \vee Q_\psi(y), \quad \text{is turned into the rule} \quad \frac{s:[\alpha]\psi,\ (s,t):\alpha}{t:\psi}.$$

With the exception of the contraction rule, all rules in Figure 6 can be obtained in this way from Figure 4. The contraction rules are the rules corresponding to factoring steps. It is not difficult to see that factoring is applicable only to hyperresolvents of an inference involving either the definitional clauses for negated conjunctions of modal formulae or those for positive occurrences of disjunctions of relational formulae.

Let Tab be the tableau calculus for $K_{(m)}(\wedge,\vee,\breve{},\rceil)$ given by the rules of Figure 4. The rules for dynamic modal formulae are the same as for $K_{(m)}$. The rules for relational formulae include two kinds of rules for every operator: an elimination rule and an introduction rule. The introduction rules are indicated by $_I$.

Lemma 5 *Tab is step-wise simulated by H_{sp} with respect to the structural transformation defined in Theorem 4 and is a reduction of it.*

Proof For step-wise simulation we need to define a function g that maps every inference step in a Tab-derivation to a sequence of H_{sp}-steps on the corresponding clauses. We just sketch how one of the relational introduction rules, the $(\wedge)^r_I$ rule, can be simulated. The unit clauses corresponding to the premises $(s,t):\alpha$ and $(s,t):\beta$ are $R_\alpha(s,t)$ and $R_\beta(s,t)$. One hyperresolution step with these and $R_{\alpha\wedge\beta}(x,y) \vee \neg R_\alpha(x,y)^+ \vee \neg R_\beta(x,y)^+$ generate the conclusion $R_{\alpha\wedge\beta}(s,t)$. The associated formula, $(s,t):\alpha\wedge\beta$, is the conclusion of the $(\wedge)^r_I$ tableau rule. It remains to check that the side conditions of the rule are satisfied in H_{sp}. The side conditions of the $(\wedge)^r_I$ rule, and the introduction rules (see Figure 6), limit the generation of formulae in the conclusions to formulae that occur in the input problem. Since the transformations of a given input problem to first-order logic and clausal form introduce new symbols only for formulae occurring in the input problem, the side conditions are satisfied in H_{sp}-derivations. In this way it is possible to show that every tableau rule can be step-wise simulated by inference steps in H_{sp}.

$$\begin{array}{lll}
(cl)\ \dfrac{s:\psi,\ s:\neg\psi}{\bot} & (\bot)\ \dfrac{s:\bot}{\bot} & \\[2ex]
(\wedge)_1\ \dfrac{s:\psi\wedge\phi}{s:\psi} & (\wedge)_2\ \dfrac{s:\psi\wedge\phi}{s:\phi} & (\neg\wedge)\ \dfrac{s:\neg(\psi\wedge\phi)}{s:{\sim}\psi \mid s:{\sim}\phi} \\[2ex]
(\neg[\cdot])_1\ \dfrac{s:\neg[\alpha]\psi}{(s,t):\alpha} & (\neg[\cdot])_2\ \dfrac{s:\neg[\alpha]\psi}{t:{\sim}\psi} & ([\cdot])\ \dfrac{s:[\alpha]\psi,\ (s,t):\alpha}{t:\psi} \\[2ex]
(\smile)\ \dfrac{(s,t):\alpha^{\smile}}{(t,s):\alpha} & (\smile)_I\ \dfrac{(t,s):\alpha}{(s,t):\alpha^{\smile}} & \\[2ex]
(\wedge)^r_1\ \dfrac{(s,t):\alpha\wedge\beta}{(s,t):\alpha} & (\wedge)^r_2\ \dfrac{(s,t):\alpha\wedge\beta}{(s,t):\beta} & (\wedge)^r_I\ \dfrac{(s,t):\alpha,\ (s,t):\beta}{(s,t):\alpha\wedge\beta} \\[2ex]
(\vee)^r\ \dfrac{(s,t):\alpha\vee\beta}{(s,t):\alpha \mid (s,t):\beta} & (\vee)^r_{I,1}\ \dfrac{(s,t):\alpha}{(s,t):\alpha\vee\beta} & (\vee)^r_{I,2}\ \dfrac{(s,t):\beta}{(s,t):\alpha\vee\beta} \\[2ex]
(\upharpoonright)_1\ \dfrac{(s,t):\alpha\upharpoonright\phi}{(s,t):\alpha} & (\upharpoonright)_2\ \dfrac{(s,t):\alpha\upharpoonright\phi}{s:\phi} & (\upharpoonright)_I\ \dfrac{(s,t):\alpha}{(s,t):\alpha\upharpoonright\phi \mid s:{\sim}\phi} \\[2ex]
(contr)\ \dfrac{s:\neg(\psi\wedge\psi)}{s:{\sim}\psi} & (contr)^r\ \dfrac{(s,t):\alpha\vee\alpha}{(s,t):\alpha} &
\end{array}$$

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 $(\smile)_I$, $(\vee)^r_{I,j}$, $(\wedge)^r_I$ and $(\upharpoonright)_I$ the side conditions are that the relational formulae in the conclusions, occur as subformulae of a box formula in the input problem.

Fig. 6 Tableau calculus for $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonright)$.

To prove that *Tab* is a reduction of H_{sp} we need to verify that the inverse mapping g^{-1} is a function. For this we need to verify that there are no inference steps in a H_{sp}-derivation for the translation of a $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonright)$-formula that are not involved in simulating tableau inference steps. This can be done by considering the possible forms of clauses in any H_{sp}-derivation and the H_{sp}-rules applicable to them. We omit the details. □

The proof shows actually something stronger:

Theorem 11 *There is a linear correspondence between Tab and H_{sp} on the structural transformation for local satisfiability in $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonright)$.*

Soundness and refutational completeness of *Tab* is now a consequence of Theorem 10.2 and Corollary 3.

Theorem 12 *A formula φ is locally satisfiable in $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonright)$ iff a tableau derivation containing a branch $\mathscr{B}$ can be constructed in Tab such that $\mathscr{B}$ does not contain $\bot$ and no more rules are applicable.*

It is possible to make the side conditions for the introduction rules of *Tab* even more restrictive and thus more efficient. This requires that we use a variation of the translation as defined in [10]. More specifically, the definition of $\mathrm{Def}(\alpha)$ needs to be varied so that different symbols are introduced for positive and negative occurrences of relational formulae. Then it follows that the introduction rules need only be applied if the formula β of the conclusion $s:\beta$ occurs as a subformula of a relational formula γ of a box formula $s{:}[\gamma]\theta$ that occurs *positively* on the current branch. The introduction rules in *Tab* are applied also for box formulae that occur negatively.

6.1 Decidability

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. For modal formulae the calculus uses only elimination rules but for relational formulae it uses both elimination rules and introduction rules. Because of the presence of the 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. Termination of the calculus follows however from the decidability of H_{sp} on the transformed problems. As a direct consequence of Theorems 8, 10.1 and 11 we obtain:

Corollary 4 *Any (fair) procedure based on Tab is a decision procedure for local satisfiability in $K_{(m)}(\wedge,\vee,{}^{\smile},\rceil)$.*

Consequently, any procedure based on Tab is a decision procedure, and no loop detection mechanism or blocking is necessary to ensure termination.

Notice the side conditions of the introduction rules follow from the clausal form of the translated problem and restrict the applications of the rules in such a way that no formulae are introduced that do not occur in the input problem. The side conditions thus imply that the calculus has the subformula property.

6.2 Finite Model Generation

The following results are extensions of results in [10, 28].

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 6 *Let φ be any $K_{(m)}(\wedge,\vee,{}^{\smile},\rceil)$-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:*

1. *I is a (Herbrand) model of N_∞ and N, if N_∞ does not contain the empty clause.*
2. *A $K_{(m)}(\wedge,\vee,{}^{\smile},\rceil)$-model of φ can be read off from I.*

Theorem 13 1. *For any modal formula locally satisfiable in $K_{(m)}(\wedge,\vee,{}^{\smile},\rceil)$ a finite model can be effectively constructed with any (fair) procedure based on H_{sp}.*
2. *$K_{(m)}(\wedge,\vee,{}^{\smile},\rceil)$ has the finite model property.*

We can state the following, due to the exact correspondence between clauses and formulae in H_{sp}- and Tab-derivations.

Corollary 5 *For any modal formula locally satisfiable in $K_{(m)}(\wedge,\vee,{}^{\smile},\rceil)$ a finite model can be effectively constructed with any (fair) procedure based on Tab.*

6.3 Redundancy in Tableaux

The ‘develop via resolution’ methodology allows us to enhance the tableau calculus with notions of redundancy that correspond to redundancy in the resolution framework. 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 $n \geq 0$) [10]. 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 (see [3]) may be chosen. For example, all tautologies in L are redundant according to this definition.

We can use the link to resolution to define a new notion of redundancy, namely *redundant inference rules*. By definition, 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. Letting $X = \emptyset$ gives redundancy of inference rules. We say an inference rule is *redundant in an calculus C*, if it is redundant with respect to $X = \emptyset$ and C. Observe that, in general, an inference rule which is not redundant in a calculus can be redundant with respect the calculus *and* some (derived) formulae.

The application of a rule is defined to be *redundant* if its conclusions are redundant in the current node or the rule is redundant (this generalises the definition found in [10, 45]). 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.

The next result gives justification for *any* instance of redundancy elimination including the examples given. It states soundness and completeness of the tableau calculus modulo redundancy. The result extends Theorem 8.1 in [10].

Theorem 14 *A formula φ is locally satisfiable in $K_{(m)}(\wedge,\vee,{}^{\smile},\upharpoonleft)$ iff a tableau derivation containing a branch $\mathscr{B}$ can be constructed in $\mathit{Tab}^{\mathrm{red}}$ (modulo redundancy) such that $\mathscr{B}$ does not contain $\bot$ and each rule application is redundant.*

Notice that this formulation of the theorem is significantly stronger and more general than the soundness and completeness results for tableaux usually found in the literature. It is not difficult to see that the Corollaries 4 and 5 hold for $\mathit{Tab}^{\mathrm{red}}$ (modulo redundancy) as well. In other words, $\mathit{Tab}^{(\mathrm{red})}$, with or without redundancy, is a decision procedure and can be used to generate finite models for $K_{(m)}(\wedge,\vee,{}^{\smile},\upharpoonleft)$.

6.4 Aside on Contraction and Factoring

Because of the presence of splitting, factoring is optional for completeness of H_{sp} on range-restricted clauses. The calculus obtained from H_{sp} without factoring is Tab without the contraction rules. It follows then that all the main results above are also true for Tab without the contraction rules (soundness, completeness, and decidability). Without contraction rules identical branches are however created and potentially considerable work is repeated during the deduction process. For reasons of efficiency it thus does make sense to extend Tab with contraction rules. Yet in the literature contraction rules are not usually included in the definition of tableau calculi or tableau procedures. However implemented tableau procedures usually include simplification steps to eliminate such and other redundancies. These simplifications are rewritings which replace obvious redundancies including duplication of the kind $\psi \wedge \psi$ and $\alpha \vee \alpha$. This makes the contraction rules superfluous.

In this paper clauses are assumed to be multisets. If clauses are assumed to be sets, then factoring is not even needed on range-restricted clauses and the mentioned inefficiency does

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

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

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

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

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

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

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

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

For the $([\cdot])_j$ rule ($j\in\{1,2\}$) the side conditions is 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 $(\upharpoonleft)_I$ the side conditions are that the complement of the relational formulae in the conclusions, occur as subformulae of a box formula in the input problem.

Fig. 7 Rasiowa-Sikorski calculus for $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonleft)$.

not arise. The methodology then produces the calculus *Tab* but without the contraction rules and the assumption that $\wedge$ and $\vee$ are idempotent operators.

7 Rasiowa-Sikorski Systems

The ‘develop via first-order resolution’ approach is not limited to the development of tableau calculi. 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 $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonleft)$ and show that it is a decision procedure and can be used for generating counter-models.

Rasiowa-Sikorski proof systems [38] are dual tableau systems [30, 49]. Given a formula F, they aim to prove its validity, or, if F is not valid, they aim to construct a counter-model, that is, 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 obvious validities such as the law of excluded middle (that is, $\neg F \vee F$). 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 a tableau derivation. There is a slight variation in notation though, and crucially the interpretation of the rules is different. 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.

A Rasiowa-Sikorski calculus for local validity in $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonleft)$ is presented in Figure 7. Let the calculus be denoted by *RS*. Notice that the rules are dual to the rules of the tableau calculus *Tab* in Figure 6. To see this, inductively define a function g such that any labelled formulae of the form $s:\psi$ or $(s,t):\alpha$ in a *Tab*-derivation is mapped to $g(s):{\sim}\psi$ and

$(g(s), g(t)) : {\sim}\alpha$ in the corresponding *RS*-derivation. If the *Tab*-derivation is started with $a : \varphi$ then the *RS*-derivation is started with $a : {\sim}\varphi$. g maps the labels as follows: $g(a) = a$ and any successor state s introduced in the $(\neg[\cdot])_j$ rules in *Tab* is mapped to the term introduced by the $([\cdot])_j$ rules in *RS*. Then extend the definition to a mapping from *Tab*-inferences (rules) to *RS*-inferences (rules).

We can show:

Theorem 15 1. *The calculi Tab and RS step-wise simulate each other in a dual manner for local satisfiability/validity in $K_{(m)}(\wedge, \vee, {}^{\smile}, \upharpoonleft)$.*

2. *There is a linear, dual correspondence between Tab and RS.*

A detailed analysis of the duality between tableau and Rasiowa-Sikorski calculi for Peirce logic which extends $K_{(m)}(\wedge, \vee, {}^{\smile}, \upharpoonleft)$ can be found in [49].

Theorem 15 implies that all the properties of the tableau calculus transfer to the Rasiowa-Sikorski calculus, and vice versa. With the notion of redundancy dualised in the expected way, we can state:

Corollary 6 *A formula φ is locally valid in $K_{(m)}(\wedge, \vee, {}^{\smile}, \upharpoonleft)$ iff a RS-derivation containing a branch $\mathscr{B}$ can be constructed (modulo redundancy, and with or without the dual contraction rules) such that $\mathscr{B}$ does not contain $\neg\bot$ and each rule application is redundant.*

Corollary 7 *Any (fair) procedure based on RS is a decision procedure for local validity in $K_{(m)}(\wedge, \vee, {}^{\smile}, \upharpoonleft)$.*

Corollary 8 *A finite modal counter-model for any modal formula which is locally invalid in $K_{(m)}(\wedge, \vee, {}^{\smile}, \upharpoonleft)$ can be effectively constructed with any (fair) procedure based on RS.*

7.1 Developing Rasiowa-Sikorski Systems via Dual Resolution

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 a legitimate 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, that is, 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 $R_{\text{sp}}^{\text{red}}$, 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 testing validity.

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 translation* of formulae in

Subformula θ	Definitional dual clauses associated with θ
(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)	$\neg R_{\neg\alpha}(x,y) \wedge \neg R_{\alpha}(x,y)$
	$R_{\neg\alpha}(x,y) \wedge R_{\alpha}(x,y)$
$\alpha \wedge \beta$	$\neg R_{\alpha\wedge\beta}(x,y) \wedge R_{\alpha}(x,y) \wedge R_{\beta}(x,y)$
	$R_{\alpha\wedge\beta}(x,y) \wedge \neg R_{\alpha}(x,y)$
	$R_{\alpha\wedge\beta}(x,y) \wedge \neg R_{\beta}(x,y)$
$\alpha \vee \beta$	$\neg R_{\alpha\vee\beta}(x,y) \wedge R_{\alpha}(x,y)$
	$\neg R_{\alpha\vee\beta}(x,y) \wedge R_{\beta}(x,y)$
	$R_{\alpha\vee\beta}(x,y) \wedge \neg R_{\alpha}(x,y) \wedge \neg R_{\beta}(x,y)$
$\alpha^{\smile}$	$\neg R_{\alpha^{\smile}}(x,y) \wedge R_{\alpha}(y,x)$
	$R_{\alpha^{\smile}}(x,y) \wedge \neg R_{\alpha}(y,x)$
$\alpha\upharpoonright\phi$	$\neg R_{\alpha\upharpoonright\phi}(x,y) \wedge R_{\alpha}(x,y) \wedge Q_{\phi}(x)$
	$R_{\alpha\upharpoonright\phi}(x,y) \wedge \neg R_{\alpha}(x,y)$
	$R_{\alpha\upharpoonright\phi}(x,y) \wedge Q_{\sim\phi}(x)$

Fig. 8 Definitional dual clausal forms for $K_{(m)}(\wedge, \vee, ^{\smile}, \upharpoonright)$

dynamic modal logic.

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

The mappings π and π' are defined as in Section 3. Here the definitions of relational formulae include the shortcut equivalence $\forall xy\,(R_{\sim\alpha}(x,y) \leftrightarrow \neg R_{\alpha}(x,y))$ which is needed to turn problems into range-restricted clauses, we discuss below.

We can prove:

Theorem 16 *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:*

1. *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 8.*
2. *φ is locally valid in L iff $\models \varphi'$ iff $\models N$.*

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

By duality, soundness and completeness (for validity) are a consequence of Corollary 1 (and also Theorem 2).

Corollary 9 *$DH^{\mathrm{red}}_{\mathrm{sp}}$ and DH_{sp} are sound and complete proof systems for sets of dual clauses.*

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

Fig. 9 Definitional dual clausal forms for $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonright)$ in range-restricted form.

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

Looking at Figure 8 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 issue can be overcome by pre-saturation and purification, and is the reason why the shortcut equivalence $\forall xy\,(R_{\sim\alpha}(x,y) \leftrightarrow \neg R_{\alpha}(x,y))$ was added to the definition of $\mathrm{Def}^{d}(\alpha)$. Take the input set N computed in accordance with Theorem 16. 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 premises we obtain the clauses in Figure 9 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 can be purified away (that is, deleted). The clauses obtained are given in Figure 9. (It is interesting to note that this partial pre-saturation corresponds to second-order quantifier elimination, see [16].)

By the way we note that alternatively, a standard shifting transformation can be used. Shifting switches the signs literals. In particular, we need to 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.

Let N' denote the 'partially pre-saturated and purified' set of clauses obtained from N as described above. Since N' is dual to the clause set obtained for satisfiability of the negated problem (compare Figures 8 and 9), we can now state:

Theorem 17 *Let φ be any $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonright)$-formula and let N' be the set of clauses obtained by the described pre-saturation and purification from the dual clausal form of the dual structural transformation. Then:*

1. *Any DH_{sp}-derivation from N' terminates.*

2. *φ is locally valid in $K_{(m)}(\wedge,\vee,{}^{\smile},\rceil)$ iff the DH_{sp}-saturation of N' contains the empty clause $\top$.*

Lemma 7 *RS is step-wise simulated by DH_{sp} with respect to set N' obtained as defined in Theorem 17, and is a reduction of it.*

Theorem 18 *There is a linear correspondence between RS and DH_{sp} on N' as defined in Theorem 17.*

Using duality and Theorems 10 and 13 we can now give alternative proofs of Corollaries 6–8 (i.e. soundness and completeness of RS, decidability and finite counter-model generation for $K_{(m)}(\wedge,\vee,{}^{\smile},\rceil)$).

8 Modal Resolution Systems

Refutation calculi without a splitting 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. These are of the kind described in Areces et al [1] for some traditional modal logics. Our approach allows us to develop such calculi also for other logics.

We focus again on the dynamic modal logic $K_{(m)}(\wedge,\vee,{}^{\smile},\rceil)$. Figure 10 presents a calculus we can read off from the structural encoding in Figure 4, considering that hyperresolution without splitting is used. C and D denote disjunctions, actually multisets, of labelled modal and relational formulae of the form $s{:}\psi$ and $(s,t){:}\alpha$. Although the labelled formulae are all unnegated, negative literals have not disappeared. We need to regard literals of the form $s{:}\neg\psi$ as negative literals. Note that there are no negative relational literals. This is because the logic $K_{(m)}(\wedge,\vee,{}^{\smile},\rceil)$ does not include an relational negation operation. Negative occurrences of relations are implicit in box formulae and are taken care of by the $([\cdot])$ rule.

Denote the calculus defined in Figure 10 as Res. For each definitional clause there is a rule in Res. In addition, there are the two factoring rules $(fact)$ and $(fact)^r$. Comparing this calculus to the tableau calculus derived earlier (that is, comparing Figure 6 and Figure 10), observe how the closure rule (cl) has become a resolution rule, the rule (res). The (res) rule is an instance of the standard, ground resolution rule. The tableau splitting rules $(\neg\wedge)$ and $(\lfloor)_I$ have become rewrite rules. With the exception of the resolution rule (res) and the factoring rules, all rules in Res are in fact rewrite rules.

Lemma 8 *Res is step-wise simulated by H (without splitting) with respect to the structural transformation defined in Theorem 4 and is a reduction of it.*

Theorem 19 *There is a linear correspondence between Res and H on the structural transformation for local satisfiability in $K_{(m)}(\wedge,\vee,{}^{\smile},\rceil)$.*

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. All inferences rules in Res, with the exception of the factoring rules, correspond to hyperresolution steps, while the factoring rules correspond to factoring on positive ground clauses in H-derivations. The correspondence between the steps is therefore slightly more direct than for tableau.

Corollary 10 *Res is sound and refutationally complete for testing the local satisfiability of formulae in $K_{(m)}(\wedge,\vee,{}^{\smile},\rceil)$.*

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

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

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

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

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

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

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

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

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.

Fig. 10 Modal resolution calculus for $K_{(m)}(\wedge,\vee,{}^{\smile},\upharpoonleft)$.

Proof By Corollary 1, Theorems 4 and Corollary 3. □

These results are an illustration that the step from tableau-style systems to systems with an explicit resolution rule is not big. The simulation results formally show that on the one hand modal resolution can be viewed as semantic tableau without splitting. On the other hand, they formally show that semantic tableau can be viewed as modal resolution with splitting. At least this is the view we get from the perspective of first-order resolution.

From the perspective of tableau and modal resolution, the essential difference between the two is the absence of splitting in modal resolution, and the difference in the rules induced by the negative propositional shortcut clause. In the case of modal resolution the negative propositional shortcut clause induces the modal resolution rule (res), whereas for tableau it induces the closure rule (because of the presence of splitting). Thus from the perspective of tableau and modal resolution the other difference besides the presence of splitting is the use of a restricted form of resolution, namely the closure rule, in the tableau calculus.

It is interesting to note that, like tableau, the modal resolution calculus uses a kind of goal-directed approach, initially breaking down the given input formula and on-the-fly transforming formulae into first-order clausal form.

One can thus view modal resolution, or semantic tableau without splitting, also as hyperresolution with lazy translation to first-order logic, because the expansion rules are just a facilitation of lazy translation to first-order logic. Similarly, semantic tableau can be viewed as hyperresolution with lazy translation to first-order logic and splitting.

8.1 Decidability

What about decidability? We can prove Theorem 10.1 also holds for hyperresolution without splitting, and any refinement of hyperresolution without splitting.

Theorem 20 *Let φ be any $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonleft)$-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 21 *Any (fair) procedure based on Res is a decision procedure for local satisfiability in $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonleft)$.*

Because we view clauses as multisets it is necessary to include the factoring rules. 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.

8.2 Refining modal resolution

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.

Let $\succ'$ be 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* (see Figure 10) but the rules have side conditions saying 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 conditions of the factoring rules are 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 10).

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

Lemma 9 *Let $\succ$ be an admissible ordering and let $\succ'$ be the corresponding ordering on labelled formulae. ORes based on $\succ'$ is step-wise simulated by OH based on $\succ$ with respect to the structural transformation defined in Theorem 4. The same statement is true for $\mathit{ORes}^{\mathrm{red}}$ and $\mathit{OH}^{\mathrm{red}}$.*

Theorem 22 *There is a linear correspondence between ORes and OH ($\mathsf{ORes}^{\mathrm{red}}$ and $\mathsf{OH}^{\mathrm{red}}$) on the structural transformation for local satisfiability in $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonleft)$.*

Theorem 23 *ORes and $\mathsf{ORes}^{\mathrm{red}}$ are sound and refutationally complete for testing local satisfiability of formulae in $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonleft)$.*

Theorem 24 *Any (fair) procedure based on $\mathsf{ORes}^{(\mathrm{red})}$ is a decision procedure for local satisfiability in $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonleft)$.*

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

The reader may be wondering how ordering restrictions can be exploited in tableau and Rasiowa-Sikorski calculi. Since splitting is always applied before hyperresolution, hyperresolution is applied to unit clauses only. This means an ordering has no effect on the application of the inference rules. Nevertheless the ordering may be used to restrict the application of splitting only to maximal literals and reduce the search space. It is beyond the scope of this paper to discuss details.

9 Discussion

It is of course 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 develop new calculi and provers. The 'develop via first-order resolution' approach provides a common framework for this which enables direct comparison of different deduction approaches and the immediate transferal of techniques between different approaches.

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 $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonleft)$ in the simulations, namely the omission of the splitting rule. Tableau and resolution methods are typically regarded as quiet opposite, with the difference being not only the presence of the splitting rule in tableau calculi but also the absence of the resolution rule in tableau calculi. Our analysis however shows (and gives a formal proof) that the difference between tableau and resolution methods is much smaller than sometimes perceived, in fact they 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 stronger soundness and completeness results and the definition of better inference systems. Our case study shows that there is significant flexibility to tailor the approach and control the kind of calculi which can be derived.

It is interesting to note that we generated all three calculi using hyperresolution. For modal resolution we also used hyperresolution and not ordinary resolution as one may have expected.

All results and observations in this paper for $K_{(m)}(\wedge,\vee,^{\smile},\upharpoonleft)$ hold for all dynamic modal logics defined over the operators $\{\wedge,\vee,^{\smile},\upharpoonleft,\downharpoonright\}$.

We know that the approach can be generalised and applied to more expressive logics to yield sound and (refutationally) complete special-purpose 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 correspondence results between tableau systems and resolution have been obtained in [19] 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 [19]. Similarly, tableau decision procedure can be defined for the solvable class $\mathscr{B}U$ introduced and studied in [18].

Soundness and completeness of the extracted calculi is in general automatic as long as the translation to first-order logic and the refinement used have the properties identified in the Introduction so that correspondence results can be proved. Apart from finding a suitable combination of a translation, refinement and pre-saturation (if needed), the main challenge is to develop ways to guarantee termination and prove decidability. In the case of $K_{(m)}(\wedge,\vee,\smile,\rceil)$ there are existing decidability results for hyperresolution which we have extended and then used to show the developed calculi provide decision procedures.

The starting point of the methodology in this paper is the semantic specification of a logic which provides the basis for the translation to first-order logic. The translation is a combination of the standard relational translation method and structural transformation. It is possible to derive in a systematic way sound and complete calculi via other, non-standard translation methods, for instance, translations based on functional translation approaches or the axiomatic translation approach. (See [37, 47] for surveys of different translation approaches and [46, 48] for the axiomatic translation.) If we use the functional translation or the optimised function translation then what we obtain are prefix tableau calculi. In prefix tableau calculi the formulae are also labelled modal formulae but the labels are strings of constants, or prefixes. With the axiomatic translation we can generate modal tableau calculi in a form that are currently popular for traditional modal logics and mainstream description logics with non-empty TBoxes. Without splitting new modal resolution calculi are produced.

It is in fact also possible to use the methodology to obtain calculi from Hilbert axiomatisations of modal logics. In [46, 48] we use resolution theorem proving techniques to obtain tableau calculi in a systematic and semi-automatic way from the Hilbert axiomatisations of traditional modal logics. Although not explicitly stated as such, the approach followed there can be seen to be an instance of 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 time and space.

There are good reasons why the methodology introduced in this paper is based on 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 developing decision procedures for non-classical logics and expressive solvable fragments of first-order logic; see the surveys [14, 29, 47]. Furthermore, the framework is well suited for developing and studying model builders; see, for instance [5, 14, 17]. 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 methodology in 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 [27, 53] and VAMPIRE [40]. 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 appropriate structural transformations 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. All these basic ingredients are already implemented in (M)SPASS [27, 53, 54].

In addition, such simulating implementations permit experimental evaluation and comparison of different deduction approaches in a uniform framework. Examples of empirical studies undertaken following essentially this approach can be found in [5, 28]. Such experiments tend to provide more reliable indicators of the suitability of different calculi, or different styles of deduction, for specific applications and logics than experiments based on the comparisons of independently implemented provers [24, 25, 28].

Further work consists of generalising the methodology and applying it to other forms of tableaux (e.g. set-labelled tableaux with global caching [21] or free variable tableaux), to other styles of deduction methods, to other logics, and other forms of inference problems. We are presently investigating the simulation and development of natural deduction calculi for modal logics and first-order logic [41]. It would also be of interest if a resolution prover can be used directly as a special-purpose prover without any extra implementation effort. Ultimately it is our aim to realise the approach as a tool that can take the specification of a logic as input and output a deduction calculus which is sound, complete and terminating, if possible. Independent student projects at Manchester have shown that it is indeed possible to use resolution theorem provers as tableau provers for propositional and modal logics. These projects use resolution theorem provers as suggested in this paper and back-translate the derived clauses into tableau derivations [22, 34]. Another student project has developed a tableau inference rule generator in Prolog [31]. We are also working on using the approach to generate tableau calculi for more expressive dynamic modal logics, in particular, our aim is to generate calculi such as the ones in [50, 51] which use only elimination rules and rules that operate only on dynamic modal formulae but not relational formulae.

Acknowledgements The work has benefitted from discussions with Ullrich Hustadt, David Robinson and Dmitry Tishkovsky. I also thank the reviewers for valuable comments. The research has been supported by UK EPSRC grants EP/D056152/1 and EP/F068530/1.

References

1. C. Areces, M. de Rijke, and H. de Nivelle. Resolution in modal, description and hybrid logic. *Journal of Logic and Computation*, 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*, chapter 2, pages 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 G. Gottlob, A. Leitsch,

and D. Mundici, editors, *Proceedings of the Third Kurt Gödel Colloquium (KGC'93)*, volume 713 of *Lecture Notes in Computer Science*, pages 83–96. Springer, 1993.

5. P. Baumgartner and R. A. Schmidt. Blocking and other enhancements for bottom-up model generation methods. In U. Furbach and N. Shankar, editors, *Automated Reasoning: Third International Joint Conference on Automated Reasoning (IJCAR 2006)*, volume 4130 of *Lecture Notes in Artificial Intelligence*, pages 125–139. Springer, 2006.
6. C. Brink, K. Britz, and R. A. Schmidt. Peirce algebras. *Formal Aspects of Computing*, 6(3):339–358, 1994.
7. 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.
8. S. A. Cook and R. A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, 1979.
9. H. De Nivelle. Splitting through new proposition symbols. In R. Nieuwenhuis and A. Voronkov, editors, *Proceedings of the Eighth International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2001)*, volume 2250 of *Lecture Notes in Artificial Intelligence*, pages 172–185. Springer, 2001.
10. H. De Nivelle, R. A. Schmidt, and U. Hustadt. Resolution-based methods for modal logics. *Logic Journal of the IGPL*, 8(3):265–292, 2000.
11. E. Eder. *Relative Complexities of First Order Calculi*. Artificial Intelligence. Vieweg, Wiesbaden, 1992.
12. L. Fariñas del Cerro and O. Gasquet. A general framework for pattern-driven modal tableaux. *Logic Journal of the IGPL*, 10(1):51–83, 2002.
13. L. Fariñas del Cerro and A. Herzig. Modal deduction with applications in epistemic and temporal logics. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming: Epistemic and Temporal Reasoning*, volume 4, pages 499–594. Clarendon Press, Oxford, 1995.
14. C. Fermüller, A. Leitsch, U. Hustadt, and T. Tammet. Resolution decision procedures. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, chapter 25, pages 1791–1849. Elsevier, 2001.
15. M. Fitting. Destructive modal resolution. *Journal of Logic and Computation*, 1(1):83–97, 1990.
16. D. M. Gabbay, R. A. Schmidt, and A. Szałas. *Second-Order Quantifier Elimination: Foundations, Computational Aspects and Applications*, volume 12 of *Studies in Logic: Mathematical Logic and Foundations*. College Publications, 2008.
17. L. Georgieva, U. Hustadt, and R. A. Schmidt. Computational space efficiency and minimal model generation for guarded formulae. In R. Nieuwenhuis and A. Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2001)*, volume 2250 of *Lecture Notes in Artificial Intelligence*, pages 85–99. Springer, 2001.
18. L. Georgieva, U. Hustadt, and R. A. Schmidt. A new clausal class decidable by hyperresolution. In A. Voronkov, editor, *Automated Deduction—CADE-18*, volume 2392 of *Lecture Notes in Artificial Intelligence*, pages 260–274. Springer, 2002.
19. L. Georgieva, U. Hustadt, and R. A. Schmidt. Hyperresolution for guarded formulae. *Journal of Symbolic Computation*, 36(1–2):163–192, 2003.
20. 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*, pages 297–396. Kluwer, 1999.
21. R. Goré and L. A. Nguyen. Exptime tableaux with global caching for description logics with transitive roles, inverse roles and role hierarchies. In N. Olivetti, editor, *Automated*

Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2007), volume 4548 of *Lecture Notes in Computer Science*, pages 133–148. Springer, 2007.

22. C. T. Gray. An automated reasoner for propositional logic. Third year project report, University of Manchester, UK, 2007.
23. I. Horrocks. *Optimising Tableaux Decision Procedures for Description Logics*. PhD thesis, University of Manchester, Manchester, UK, 1997.
24. U. Hustadt and R. A. Schmidt. An empirical analysis of modal theorem provers. *Journal of Applied Non-Classical Logics*, 9(4):479–522, 1999.
25. U. Hustadt and R. A. Schmidt. On the relation of resolution and tableaux proof systems for description logics. In T. Dean, editor, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI'99)*, volume 1, pages 110–115. Morgan Kaufmann, 1999.
26. U. Hustadt and R. A. Schmidt. Issues of decidability for description logics in the framework of resolution. In R. Caferra and G. Salzer, editors, *Automated Deduction in Classical and Non-Classical Logics*, volume 1761 of *Lecture Notes in Artificial Intelligence*, pages 191–205. Springer, 2000.
27. U. Hustadt and R. A. Schmidt. MSPASS: Modal reasoning by translation and first-order resolution. In R. Dyckhoff, editor, *Automated Reasoning with Analytic Tableaux and Related Methods, International Conference (TABLEAUX 2000)*, volume 1847 of *Lecture Notes in Artificial Intelligence*, pages 67–71. Springer, 2000.
28. U. Hustadt and R. A. Schmidt. Using resolution for testing modal satisfiability and building models. In I. P. Gent, H. van Maaren, and T. Walsh, editors, *SAT 2000: Highlights of Satisfiability Research in the Year 2000*, volume 63 of *Frontiers in Artificial Intelligence and Applications*, pages 459–483. IOS Press, Amsterdam, 2000. Also published as [**?**].
29. U. Hustadt, R. A. Schmidt, and L. Georgieva. A survey of decidable first-order fragments and description logics. *Journal of Relational Methods in Computer Science*, 1:251–276, 2004.
30. B. Konikowska. Rasiowa-Sikorski deduction systems in computer science applications. *Theoretical Computer Science*, 286(2):323–366, 2002.
31. C. Kourouyiannis. A tableau calculus generator for modal logic. Master's thesis, University of Manchester, UK, 2007.
32. F. Massacci. Single step tableaux for modal logics: Computational properties, complexity and methodology. *Journal of Automated Reasoning*, 24(3):319–364, 2000.
33. F. Massacci. Decision procedures for expressive description logics with intersection, composition, converse of roles and role identity. In B. Nebel, editor, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI 2001)*, pages 193–198. Morgan Kaufmann, 2001.
34. S. M. Mostafavi-Taraghi. Resolution-based tableaux for modal and description logics. Master's thesis, University of Manchester, UK, 2007.
35. K. Nellas. Reasoning about sets and relations: A tableaux-based automated theorem prover for Peirce logic. Master's thesis, University of Manchester, UK, 2001.
36. R. Nieuwenhuis and A. Rubio. Paramodulation-based theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, pages 371–443. Elsevier, 2001.
37. H. J. Ohlbach, A. Nonnengart, M. de Rijke, and D. Gabbay. Encoding two-valued non-classical logics in classical logic. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, chapter 21, pages 1403–1486. Elsevier Science, 2001.

38. H. Rasiowa and R. Sikorski. *The Mathematics of Metamathematics*, volume 41 of *Monografie matematyczne*. Polish Scientific Publ., Warsaw, 1963.
39. A. Riazanov and A. Voronkov. Splitting without backtracking. In B. Nebel, editor, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI 2001)*, pages 611–617. Morgan Kaufmann, 2001.
40. A. Riazanov and A. Voronkov. The design and implementation of VAMPIRE. *AI Communications*, 15(2-3):91–110, 2002.
41. D. Robinson and R. A. Schmidt. Resolution and natural deduction. In preparation, 2007.
42. J. A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, 1965.
43. R. A. Schmidt. Developing modal tableaux and resolution methods via first-order resolution. In G. Governatori, I. Hodkinson, and Y. Venema, editors, *Advances in Modal Logic, Volume 6*, pages 1–26, London, 2006. College Publications.
44. R. A. Schmidt and U. Hustadt. A resolution decision procedure for fluted logic. In D. McAllester, editor, *Automated Deduction—CADE-17*, volume 1831 of *Lecture Notes in Artificial Intelligence*, pages 433–448. Springer, 2000.
45. R. A. Schmidt and U. Hustadt. Mechanised reasoning and model generation for extended modal logics. In H. C. M. de Swart, E. Orlowska, G. Schmidt, and M. Roubens, editors, *Theory and Applications of Relational Structures as Knowledge Instruments*, volume 2929 of *Lecture Notes in Computer Science*, pages 38–67. Springer, 2003.
46. R. A. Schmidt and U. Hustadt. A principle for incorporating axioms into the first-order translation of modal formulae. In F. Baader, editor, *Automated Deduction—CADE-19*, volume 2741 of *Lecture Notes in Artificial Intelligence*, pages 412–426. Springer, 2003. The extended, long version is [48].
47. R. A. Schmidt and U. Hustadt. First-order resolution methods for modal logics. In *Volume in memoriam of Harald Ganzinger*, Lecture Notes in Computer Science. Springer, 2006. To appear.
48. R. A. Schmidt and U. Hustadt. The axiomatic translation principle for modal logic. *ACM Transactions on Computational Logic*, 8(4):1–55, 2007.
49. R. A. Schmidt, E. Orlowska, and U. Hustadt. Two proof systems for Peirce algebras. In R. Berghammer, B. Möller, and G. Struth, editors, *Relational and Kleene-Algebraic Methods in Computer Science (RelMiCS 7)*, volume 3051 of *Lecture Notes in Computer Science*, pages 238–251. Springer, 2004.
50. R. A. Schmidt and D. Tishkovsky. Using tableau to decide expressive description logics with role negation. In K. Aberer, K.-S. Choi, N. Fridman Noy, D. Allemang, K.-I. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, editors, *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11–15, 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 438–451. Springer, 2007.
51. R. A. Schmidt and D. Tishkovsky. A general tableau method for deciding description logics, modal logics and related first-order fragments. In A. Armando, P. Baumgartner, and G. Dowek, editors, *Automated Reasoning (IJCAR 2008)*, volume 5195 of *Lecture Notes in Computer Science*, pages 194–209. Springer, 2008.
52. S. Schulz. E: A brainiac theorem prover. *AI Communications*, 15(2–3):111–126, 2002.
53. C. Weidenbach, R. A. Schmidt, T. Hillenbrand, R. Rusev, and D. Topic. System description: SPASS version 3.0. In F. Pfenning, editor, *Automated Deduction—CADE-21*, volume 4603 of *Lecture Notes in Artificial Intelligence*, pages 514–520. Springer, 2007.

54. C. Weidenbach, R. A. Schmidt, and E. Keen. SPASS handbook version 3.0. Contained in the distribution of SPASS Version 3.0, 2007.