

A Tableau Calculus for Minimal Modal Model Generation

Fabio Papacchini¹ and Renate A. Schmidt²

School of Computer Science, University of Manchester

Abstract

Model generation and minimal model generation is useful for fault analysis, verification of systems and validation of data models. Whereas for classical propositional and first-order logic several model minimization approaches have been developed and studied, for non-classical logic the topic has been much less studied. In this paper we introduce a minimal model generation calculus for multi-modal logic $\mathbf{K}_{(m)}$ and extensions of $\mathbf{K}_{(m)}$ with the axioms **T** and **B**. The calculus provides a method to generate all and only minimal modal Herbrand models, and each model is generated exactly once. A novelty of the calculus is a non-standard complement splitting rule designed for minimal model generation. Experiments show the rule has the added benefit of reducing the search space.

Keywords: tableaux calculus, modal logic, minimal model generation, model generation

1 Introduction

Model generation and minimal model generation is useful for fault analysis, verification of systems and validation of data models ([16,1]). For classical propositional and first-order logic several approaches have been developed for model minimization. These existing approaches can be classified as belonging to three different categories: those aiming to minimize the domain of interpretation (for example [8,10]), those aiming to minimize the interpretation of certain predicates (for example [11,12]), and those aiming to minimize the interpretation of all predicates (for example [3,13,5]).

For modal logics and related description logics minimal model generation has not been studied much. Minimal model generation has received most attention for modal logics with non-monotonic operators and non-monotonic semantics, where the aim is the minimization of certain predicates (for example [6,7]). As the common modal logics can be translated into first-order logic [14], classical approaches for

¹ Email: papacchf@cs.man.ac.uk

² Email: schmidt@cs.man.ac.uk

minimal model generation can be used to generate minimal models for modal logic formulae by using a translation approach. This approach is taken in [5], which is based on earlier work for using hyperresolution to generate Herbrand models for modal problems [9,4] and [3,13].

In this paper we focus on *the generation of minimal Herbrand models*. Though minimal Herbrand models are not domain minimal, in certain applications they tend to be more natural than domain minimal models. For example, a domain minimal model of the labelled modal formula $Bob : \langle has_father \rangle doctor$ is $\{Bob : doctor, (Bob, Bob) : has_father\}$. This says that Bob is his own father. In contrast, the minimal Herbrand model is $\{f_{\langle has_father \rangle doctor}(Bob) : doctor, (Bob, f_{\langle has_father \rangle doctor}(Bob)) : has_father\}$, where Bob 's father is represented by the Skolem term $f_{\langle has_father \rangle doctor}(Bob)$ respecting the more natural meaning of the has_father relation.

We introduce a *modal* approach to minimal Herbrand model generation for the multi-modal logic $\mathbf{K}_{(m)}$ and its extensions with axioms **T** and **B**, that are represented by reflexive and symmetric accessibility relations. While being inspired by the PUHR approach [3] for first-order logic, our approach is based on a standard semantic labelled tableau calculus that has been adapted for generating minimal models. The calculus is designed so that the models induced by any fully expanded, open tableau branch are minimal.

Our calculus is called the *3MG calculus*, where 3MG is short for ‘minimal modal model generation’. Rather than using an explicit analytic cut rule and testing minimality by a second application of (a variation of) the calculus, as is done for example in [13,12,5,6,7], the idea of the 3MG calculus is to use complement splitting and model constraint propagation during backtracking to generate minimal models. The calculus takes as input a set of tableau clauses and returns in one run all minimal modal Herbrand models. Tableau clauses are disjunctions of labelled modal formulae and labelled relations. All models that are generated are minimal and no model is generated more than once.

The paper is structured as follows. We define the syntax and semantics and all important notions of our tableau language in Section 2, where we also give the formal definition of (minimal) modal Herbrand model. In Section 3 the 3MG calculus is defined and the generation of minimal models is illustrated with two examples. We summarize the proof of minimal model soundness and completeness in Section 4. (A detailed exposition of the proof can be found in the long version of the paper [15].) We conclude the paper with a discussion of related work and practical benefits of our non-standard complement splitting rule (Section 5), and a short summary and outlook (Section 6).

2 Tableau Language

Our tableau calculus is designed for sets of modal formulae of propositional multi-modal logic in which the modal operators are **K**-modalities, **T**-modalities, **B**-modalities and **TB**-modalities. Semantically these modalities are characterized

Table 1
Modalities and their corresponding frame conditions

$[R_i]$	Axiom	Frame condition
K		
KT	$[R_i]p \rightarrow p$	reflexivity
KB	$p \rightarrow [R_i]\langle R_i \rangle p$	symmetry
KTB	$[R_i]p \rightarrow p$	reflexivity
	$p \rightarrow [R_i]\langle R_i \rangle p$	symmetry

by no frame condition, reflexivity, symmetry and both reflexivity and symmetry, as indicated in Table 1.

A *modal formula* is a formula of the form \top , \perp , p_i , $\neg\phi$, $\phi_1 \wedge \phi_2$, $\phi_1 \vee \phi_2$, $\langle R_i \rangle \phi$, $[R_i]\phi$, where \top and \perp are two nullary logical operators for, respectively, true and false; p_i is a propositional symbol; R_i denotes an accessibility relation; \neg , \wedge , \vee , $\langle R_i \rangle$, $[R_i]$ are, respectively, the logical operators negation, conjunction, disjunction, diamond and box; and ϕ_i is a modal formula.

A subformula ϕ' of a modal formula ϕ has *positive polarity* if ϕ' is (implicitly or explicitly) in the scope of an even number of negations. A subformula ϕ' of a modal formula ϕ has *negative polarity* if ϕ' is (implicitly or explicitly) in the scope of an odd number of negations. A modal formula $\sim\phi$ is defined as ϕ_1 if $\phi = \neg\phi_1$, and $\neg\phi_1$ otherwise.

The tableau calculus operates on *tableau clauses*, which are disjunctions of labelled modal formulae and labelled relations. A *labelled (modal) formula* is a pair $u : \phi$ where u is a label, ϕ is a multi-modal formula and the components of the pair are divided by the operator $:$. The operator $:$ is assumed to have priority over all other operators. The labels are *terms* built from a supply of constants and unary function symbols. Intuitively, $u : \phi$ means that ϕ is true in the world represented by the term u . A *labelled relation* R_i is either of the form $(u, v) : R_i$ or $(u, v) : \neg R_i$, where u and v are terms. Intuitively, $(u, v) : R_i$ means that there is a relation R_i between u and v , while $(u, v) : \neg R_i$ means that there is no relation R_i between u and v .

Formally, *tableau clauses* are defined by the following Backus-Naur Form production rule:

$$TC ::= \top \mid \perp \mid u : \phi \mid (u, v) : R_i \mid (u, v) : \neg R_i \mid TC \vee TC.$$

A *positive tableau literal* is a tableau clause of the form $u : p_i$, $u : \langle R_i \rangle \phi$ or a positive labelled relation $(u, v) : R_i$. A *negative tableau literal* is a tableau clause of the form $u : \neg p_i$, $u : [R_i]\phi$ or a negative labelled relation $(u, v) : \neg R_i$. A *tableau atom* is a positive tableau literal of the form $u : p_i$ or $(u, v) : R_i$. We use the symbol \mathcal{P} for positive tableau literals, the symbol Δ for tableau clauses, and Δ^+ for tableau clauses consisting only of positive tableau literals.

As our aim is to generate Herbrand models, we focus our attention on defining the notions of modal Herbrand interpretation and modal Herbrand model. It is however not difficult to extend the definition to the more general case, and showing through a specialization of the Herbrand theorem that each modal Herbrand

Table 2
Semantics of tableau formulae. Note: $u, v \in W_{\mathcal{U}}$, ϕ and ϕ_i denote modal formulae, Δ_i denotes tableau clauses

$I \models u : p_i$ iff $u : p_i \in I$	$I \models (u, v) : R_i$ iff $(u, v) : R_i \in I$
$I \not\models \perp$	$I \models \top$
$I \not\models u : \perp$	$I \models u : \top$
$I \models u : \neg\phi$ iff $I \not\models u : \phi$	$I \models (u, v) : \neg R_i$ iff $I \not\models (u, v) : R_i$
$I \models u : (\phi_1 \vee \phi_2)$ iff $I \models u : \phi_1$ or $I \models u : \phi_2$	$I \models \Delta_1 \vee \Delta_2$ iff $I \models \Delta_1$ or $I \models \Delta_2$
$I \models u : [R_i]\phi$ iff for every v if $(u, v) : R_i \in I$ then $I \models v : \phi$	
$I \models u : \langle R_i \rangle \phi$ iff $(u, f_{\langle R_i \rangle \phi}(u)) : R \in I$ and $I \models f_{\langle R_i \rangle \phi}(u) : \phi$	

interpretation is a standard interpretation.

Given a set of tableau clauses N , let $W_{\mathcal{U}}$ be the set of all terms built from a supply of unary function symbols of the form $f_{\langle R_i \rangle \phi_i}$ and $f_{[R_i] \sim \phi_i}$, and the terms appearing in N . The notation indicates that $f_{\langle R_i \rangle \phi_i}$ is uniquely associated with subformulae $\langle R_i \rangle \phi_i$ of a labelled ϕ in N with positive polarity, and $f_{[R_i] \sim \phi_i}$ is uniquely associated with subformulae $[R_i] \phi_i$ of a labelled ϕ in N with negative polarity. The set $W_{\mathcal{U}}$ is the *modal Herbrand universe* for N .

The modal Herbrand semantics of tableau clauses is given by a modal Herbrand interpretation I . A *modal Herbrand interpretation* I for a tableau clause Δ is a possibly empty set of positive tableau atoms, with all terms occurring in it belonging to $W_{\mathcal{U}}$. Truth in a modal Herbrand interpretation I is inductively defined in Table 2.

If a set of tableau clauses N is true in a modal Herbrand interpretation I then I is said to be a *modal Herbrand model* for N .

A property that follows directly from the definition is the following. For any interpretation I ,

$$I \models u : (\phi_1 \vee \phi_2) \text{ iff } I \models u : \phi_1 \vee u : \phi_2. \quad (1)$$

Herbrand interpretations as defined above can be conveniently ordered by the subset relation. Let I and I' be two modal Herbrand interpretations. If $I \subseteq I'$, then we write $I \leq I'$. Given a set of tableau clauses N and a modal Herbrand model I of N , I is a *minimal modal Herbrand model* of N iff for every other modal Herbrand model I' of N , if $I' \leq I$ then $I = I'$.

For example, the minimal modal Herbrand models for the tableau clause $w : (p_1 \wedge (\langle R_1 \rangle p_2 \vee p_3))$ in a multi-modal logic $\mathbf{K}_{(m)}$ frame are $I_1 = \{w : p_1, w : p_3\}$ and $I_2 = \{w : p_1, f_{\langle R_1 \rangle p_2}(w) : p_2, (w, f_{\langle R_1 \rangle p_2}(w)) : R_1\}$. In this case $I_3 = \{w : p_1, w : p_3, f_{\langle R_1 \rangle p_2}(w) : p_2, (w, f_{\langle R_1 \rangle p_2}(w)) : R_1\}$ is also a modal Herbrand model of the tableau clause under consideration, but I_3 is not minimal, because it is a supermodel of at least one of the other models, in fact, of both of them.

3 Minimal Modal Model Generation Calculus

The input of the 3MG calculus is a set of tableau clauses such that conjunction appears in a modal formula only in the scope of a diamond operator.

Given a set of labelled modal formulae, to obtain the required input we apply a clausal normal form transformation to the labelled modal formulae in the usual way, with the addition of box miniscoping. Box miniscoping is the exhaustive application of the rule $[R_i](\phi_1 \wedge \phi_2) \Rightarrow [R_i]\phi_1 \wedge [R_i]\phi_2$, that is, the box operator is distributed as

Table 3

The rules of the 3MG calculus. Note: \mathcal{P} denotes any positive tableau literal, Δ denotes any tableau clause, Δ^+ denotes any disjunction of positive tableau literals

Expansion rules	
$(\mathbf{T})^i \frac{}{(u, u) : R_i}$ <p>if R_i is reflexive and u appears in a tableau formula of the form $u : \phi$, $(u, v) : R_j$ or $(v, u) : R_j$ on the current branch</p>	$(\mathbf{B})^i \frac{(u, v) : R_i}{(v, u) : R_i}$ <p>if R_i is symmetric</p>
$(\diamond) \frac{u : \langle R_i \rangle (\phi_1 \wedge \dots \wedge \phi_n)}{(u, f_{\langle R_i \rangle \phi}(u)) : R_i}$ $f_{\langle R_i \rangle \phi}(u) : \phi_1$ \vdots $f_{\langle R_i \rangle \phi}(u) : \phi_n$ <p>where $\phi = \phi_1 \wedge \dots \wedge \phi_n$ and $f_{\langle R_i \rangle \phi}$ is the function symbol uniquely associated with $\langle R_i \rangle \phi$</p>	$(\vee)_E \frac{u : (\phi_1 \vee \dots \vee \phi_n) \vee \Delta}{u : \phi_1 \vee \dots \vee u : \phi_n \vee \Delta}$ $(CS) \frac{\mathcal{P}_1 \vee \dots \vee \mathcal{P}_n}{\mathcal{P}_1 \mid \mathcal{P}_2 \vee \dots \vee \mathcal{P}_n}$ <p>where $neg(\mathcal{P}_i)$ stands for $neg(\mathcal{P}_2), \dots, neg(\mathcal{P}_n)$</p>
$(SBR) \frac{\begin{array}{c} u_1 : p_1 \quad \dots \quad u_n : p_n \\ (v_1, w_1) : R_{m_1} \quad \dots \quad (v_m, w_m) : R_{m_m} \\ (s_1, t_1) : R_{j_1} \quad \dots \quad (s_j, t_j) : R_{j_j} \\ u_1 : \neg p_1 \vee \dots \vee u_n : \neg p_n \vee v_1 : [R_{m_1}] \phi_1 \vee \dots \vee v_m : [R_{m_m}] \phi_m \\ \vee (s_1, t_1) : \neg R_{j_1} \vee \dots \vee (s_j, t_j) : \neg R_{j_j} \vee \Delta^+ \end{array}}{(w_1 : \phi_1) \vee \dots \vee (w_m : \phi_m) \vee \Delta^+}$	

Model constraint propagation rule

If \mathcal{B} is an open and fully expanded branch in a tableau derivation generated by the 3MG calculus, and $I = \{u_1 : p_1, \dots, u_n : p_n, (v_1, w_1) : R_1, \dots, (v_m, w_m) : R_m\}$ is the (minimal) modal Herbrand model extracted from \mathcal{B} , then the following *model constraint clause*

$$u_1 : \neg p_1 \vee \dots \vee u_n : \neg p_n \vee (v_1, w_1) : \neg R_1 \vee \dots \vee (v_m, w_m) : \neg R_m$$

is added to all the branches to the right of \mathcal{B} .

far as possible over conjunctions. This ensures that in a modal formula a conjunction may appear only in the scope of a diamond operator, not a box operator.

For example, consider the labelled formula $w : (p_2 \vee [R_1](p_1 \vee (p_2 \wedge \langle R_2 \rangle (p_1 \vee (p_2 \wedge p_3))))$. Its conjunctive normal form is (2), and the input to the calculus is the set (3).

$$w : (p_2 \vee [R_1](p_1 \vee p_2)) \wedge (p_2 \vee [R_1](p_1 \vee \langle R_2 \rangle ((p_1 \vee p_2) \wedge (p_1 \vee p_3)))) \quad (2)$$

$$\{ w : (p_2 \vee [R_1](p_1 \vee p_2)), \\ w : (p_2 \vee [R_1](p_1 \vee \langle R_2 \rangle ((p_1 \vee p_2) \wedge (p_1 \vee p_3)))) \} \quad (3)$$

The 3MG calculus consists of the six expansion rules and the model constraint propagation rule listed in Table 3.

The $(\mathbf{T})^i$ rule accommodates the \mathbf{T} axiom in the calculus, that is, it expresses the reflexivity property for relations that are known to be reflexive. The rule is necessarily different from the rule commonly used in other tableau calculi, because terms appearing in a clause generated by the model constraint propagation rule or the negation of a diamond formula may not appear in any other tableau literals. In

this case the $(\mathbf{T})^i$ rule does not have to create any relation on them, as shown in the second example at the end of this section.

The $(\mathbf{B})^i$ rule is the standard structural rule for accommodating the frame condition for \mathbf{B} .

The (\diamond) rule is the union of the standard α rule for conjunctive formulae and the diamond rule of standard multi-modal tableaux calculi. No separate α rule is needed since formulae in the input set and derived formulae are in a normal form where conjunctions can appear only immediately below a diamond operator. Another important difference to common definitions found in the literature is that the diamond rule does not create a new constant, but a new Skolem term of the form $f_{\langle R_i \rangle \phi}(u)$.

Since the other rules of the calculus are applicable only to disjunctions of tableau literals, the $(\vee)_E$ rule converts disjunctions of modal formulae under a specific label to disjunctions of labelled literals. The $(\vee)_E$ rule is the only rule that does not contribute to the generated model, since it does not add any positive or negative tableau literals to the branch. The rule is justified by the property (1) in Section 2.

The (CS) rule is the complement splitting rule. Its premise is a disjunction of positive tableau literals. An application of the (CS) rule results in the creation of two branches. One of the positive tableau literals in the premise and the negation of all the other literals are added to the left branch. The premise, with the positive tableau literal appearing on the left branch removed, is added to the right branch. Here, the negation of literals is defined by a unary function *neg* as follows.

$$\text{neg}(\mathcal{P}) = \begin{cases} u : \neg p_i & \text{if } \mathcal{P} = u : p_i \\ (u, v) : \neg R_i & \text{if } \mathcal{P} = (u, v) : R_i \\ (u, f_{\langle R_i \rangle \phi}(u)) : \neg R_i & \text{if } \mathcal{P} = u : \langle R_i \rangle \phi. \end{cases}$$

That is, if a positive tableau literal has the form $u : p_i$ or $(u, v) : R_i$, then its negation is simply $u : \neg p_i$ or $(u, v) : \neg R_i$, respectively. The negation of positive tableau literals of the form $u : \langle R_i \rangle \phi$ needs special handling. It is not possible to negate the diamond formula as might be expected, because this could produce non-minimal models (in Section 5 we give an example). Instead we define the negation of $u : \langle R_i \rangle \phi$ to be $(u, f_{\langle R_i \rangle \phi}(u)) : \neg R_i$. The intuition is that if there is a negation of a positive tableau literal, then we want to avoid the presence and the expansion of that positive literal in this branch. Following the modal Herbrand interpretation semantics, to block a specific diamond formula we can use the relation that such a diamond formula would create if expanded. For this to work it is important that the relation is uniquely associated to that diamond formula via the terms, which is achieved in our calculus through the use of Skolem functions in the way defined.

The (CS) rule is the only branching rule of the calculus, and its aim is twofold. First, it avoids the creation of a model more than once, because each branch differs from any other branch by at least one model element (tableau atom). Second, the first model extracted from the left-most branch of the tableau is minimal. These two properties are consequences of the soundness and completeness result sketched

in Section 4.

The last expansion rule is what we refer to as the (*SBR*) rule. The name reflects the close relationship to selection-based resolution for first-order clause logic. The (*SBR*) rule is the most complex rule and is the only rule that can close a branch. It may be thought of as the simultaneous application of closure rules (for labelled formulae and labelled relations) and the box rule in standard multi-modal tableau calculi. The aim of the (*SBR*) rule is to expand a disjunction of tableau literals where some of the tableau literals are negative iff it is necessary. This behaviour is based on our definition of minimal modal Herbrand models, in fact, such models are composed only of specific positive tableau literals. Thus, if the expansion of a tableau clause results in a clause that contains at least one negative literal then such a clause does not contribute to the model. As the box operator hides complex modal formulae, the rule does not completely avoid the generation of clauses that may contain negative tableau literals, but it tries to avoid them as much as possible (cf. Section 5).

The last rule in the calculus is the model constraint propagation rule. It is different from the other rules in that it becomes applicable once a fully expanded, open branch has been obtained. A branch is *fully expanded* if no more rules are applicable. If the current branch \mathcal{B} is open and fully expanded, then the model constraint propagation rule extracts the Herbrand model defined by the positive tableau atoms in \mathcal{B} . This Herbrand model is used to construct the model constraint clause as described in Table 3. The modal constraint clause is added to all branches to the right of the current branch. The calculus is defined in such a way—and derivations are constructed in such a way—that any model extracted from a fully expanded, open branch is a minimal Herbrand model. The minimal model constraints added during backtracking prevent the generation of non-minimal models by immediately closing branches which begin to construct super-models. If a super-model of an already extracted model is constructed in a branch, then an application of the (*SBR*) rule with the model constraint clause as main premise closes the branch.

We assume as usual that no rule is applied more than once to the same set of premises.

The 3MG calculus is *minimal model sound and complete*, in the sense that it terminates and generates all and only minimal modal Herbrand models for a set of tableau clauses. These properties of the calculus are not only due to its rules, but also due to the search strategy used during the derivation. We assume that a depth-first left-to-right expansion strategy is used. A departure from this strategy would compromise minimal model soundness and completeness of the calculus.

The expansion rules may be applied in any order without compromising minimal model soundness and completeness. A sensible order of application is: $(\mathbf{T})^i$, $(\mathbf{B})^i$, (*SBR*), $(\vee)_E$, (\diamond) , and (*CS*), the idea being to close a branch as soon as possible to avoid useless expansion, and to delay the application of the branching rule to avoid repeated application of a rule in different branches.

For a given input set N of tableau clauses the 3MG calculus derives either a closed tableau or a fully expanded, open tableau. If a closed tableau is con-

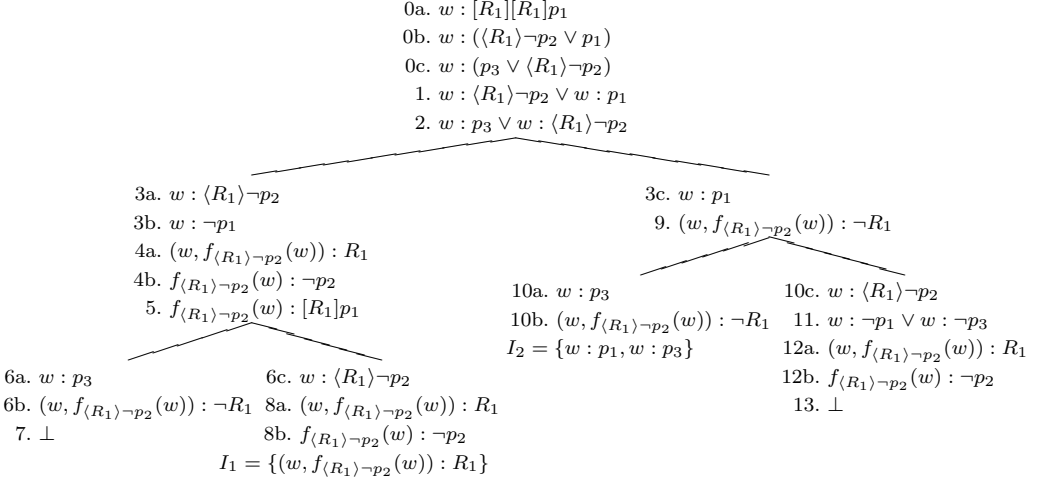


Fig. 1. Derivation for the set $\{w : [R_1][R_1]p_1, w : (\langle R_1 \rangle \neg p_2 \vee p_1), w : (p_3 \vee \langle R_1 \rangle \neg p_2)\}$ in the multi-modal $\mathbf{K}_{(m)}$ frame. The models returned are $I_1 = \{(w, f_{\langle R_1 \rangle \neg p_2}(w)) : R_1\}$ and $I_2 = \{w : p_1, w : p_3\}$

structed, N is unsatisfiable. If a fully expanded, open tableau is constructed, N is satisfiable, and each open branch defines a minimal modal Herbrand model. As the 3MG calculus uses a depth-first left-to-right strategy, the process could be stopped after the first fully expanded, open branch has been constructed, if we are interested in finding only one minimal model.

We conclude this section with two examples. First, the set $\{w : [R_1][R_1]p_1, w : (\langle R_1 \rangle \neg p_2 \vee p_1), w : (p_3 \vee \langle R_1 \rangle \neg p_2)\}$ is $\mathbf{K}_{(m)}$ -satisfiable and has two minimal Herbrand models. Figure 1 shows how these can be derived using our tableau calculus. Each formula in the derivation is numbered, the convention being that each number represents the application of a rule. The number 0 identifies the input clauses. In this example at least one application of each rule is shown, with the exception of the rules representing the **T** and **B** axioms. The $(\vee)_E$ rule is applied to 0b and 0c to derive 1 and 2, to which the complement splitting rule is now applicable. The formulae numbered 3 are obtained by applying (CS) to 1. The (\diamond) rule is applied to 3a, 6c and 10c to respectively get 4, 8 and 12. As the example is simple, all (\diamond) rule applications are equivalent to applying the standard diamond rule modulo Skolem terms being introduced, which is one of the features of the calculus. The derivation shows different applications of the (CS) rule. It is possible to observe the function *neg* in operation for a diamond formula in the formulae numbered 6 and 10 that are the result of applying the (CS) rule to 2. In this case we see the *neg* function blocks the expansion of the positive tableau literal in the input. In particular, the branch finishing with 7 closes due to the contradiction between 6b and the already expanded diamond (represented by 4). The branch closes as a result of the application of the (SBR) rule to 6b and 4a. During the explanation of the (SBR) rule, we pointed out that an application of the (SBR) rule may lead to a tableau formula containing negative tableau literals. An example of this is formula 5, which is the result of applying the (SBR) rule to 0a and 4a. As the branch ending with 8

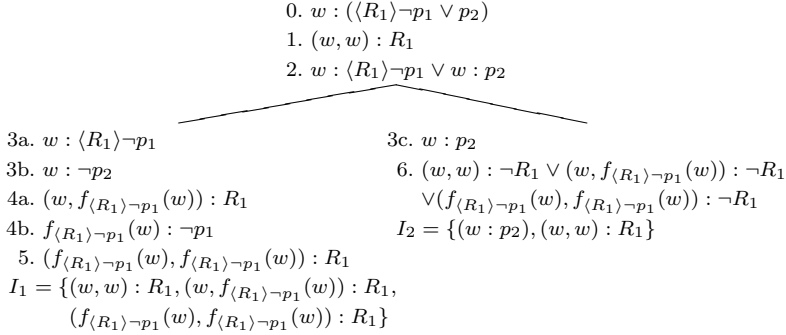


Fig. 2. Derivation for the tableau clause $w : (\langle R_1 \rangle \neg p_1 \vee p_2)$ and R_1 reflexive.

is open and fully expanded, the minimal modal Herbrand model that is given is extracted. The model constraint generated from this model is added to the only other branch as model constraint 9. In the right-most branch it is possible to note how the model constraint avoids the creation of super-models. In fact, the branch is closed by an application of the (*SBR*) rule to 9 and 12a.

The second example is shown in Figure 2. The input set is composed of the single tableau clause $w : (\langle R_1 \rangle \neg p_1 \vee p_2)$ and R_1 is assumed to be reflexive. Thanks to the side conditions of the (**T**)ⁱ rule, the relation $(f_{\langle R_1 \rangle \neg p_1}(w), f_{\langle R_1 \rangle \neg p_1}(w)) : R_1$ does not appear in the right branch, because the term $f_{\langle R_1 \rangle \neg p_1}(w)$ appears only in the model constraint clause (6). A standard rule for the **T** axiom would add the relation to the right branch, resulting in a non-minimal model.

4 Soundness and Completeness

For lack of space in this section we only give the general idea of the minimal model soundness and completeness proof of the 3MG calculus. A more detailed proof can be found in [15].

Our proof is based on showing that there exists a bisimulation between our calculus and the *depth-first minimal model generation procedure* presented in [3], called the PUHR (positive unit hyperresolution) approach, applied to a slight modification of the relational translation of the multi-modal logic to the first-order logic ([14]). The proof consists of three steps: a specialised translation from multi-modal formulae to first-order formulae, a variation of the PUHR approach preserving the minimal model soundness and completeness, and proving the equivalence between our calculus and the variation of the PUHR approach. Additionally we show termination for the 3MG calculus.

The modification of the relational translation is the application of a limited form of structural transformation which introduces a new unary predicate symbol for each labelled modal formula $u : \phi$ occurring in the input set of tableau clauses to the 3MG calculus. Together with inner Skolemization this implies that a unique Skolem function symbol is assigned to the translation of each distinct diamond. Consequently, the constants in the first-order translation are the original terms of the 3MG input. This modification to the relational translation is an essential step,

else the PUHR approach can generate non-minimal models for the original problem set.

The PUHR approach requires as input a set of range-restricted first-order clauses and that the Herbrand models of the input are all finite. Our translation from modal formulae to first-order formulae matches the requirements of the PUHR approach and preserves minimal Herbrand models. The main features of our translation are a new translation of diamond formulae and a non-standard first-order logic encoding of the reflexivity frame correspondence property. The new translation of diamond formulae, which we call *diamond optimization transformation*, allows us to have a first-order logic explanation of the use of the *neg* function instead of negating diamond formulae. It is a crucial step in the equivalence proof. As the reflexivity clause $R(x, x)$ is not range-restricted, a special encoding of reflexivity in terms of range-restricted clauses is needed (it happens to be an encoding of the $(\mathbf{T})^i$ rule with its side-conditions).

The next step of the proof is the substitution of the positive unit hyperresolution rule with a selection based resolution rule where the side premises are positive ground unit clauses. We refer to this modification as *the SBR approach*. Like the PUHR approach, the SBR approach is minimal model sound and complete.

Finally, it can be shown that there is a minimal Herbrand model preserving bisimulation relationship between the SBR approach and our 3MG calculus.

This proves our main result:

Theorem 4.1 *Let N be a set of tableau clauses. The 3MG calculus applied to N terminates, generates all and only minimal modal Herbrand models of N , and each model is generated no more than once.*

5 Discussion

The presented 3MG calculus generates all and only minimal modal Herbrand models working directly with modal formulae. To our knowledge there are no other methods having the same aim and working directly with modal formulae.

The closest approach that aims to generate all and only minimal Herbrand models for modal logic is described in [5]. The approach in [5] covers the GF^- fragment of first-order logic that is wider than the multi-modal logic $\mathbf{K}_{(m)}$ and extensions with axioms \mathbf{T} and \mathbf{B} . However, the approach does not operate on modal formulae but on their translation into first-order logic. Its results differ slightly from the results obtained using the 3MG calculus. Specifically, the set of models generated by our calculus is a subset of the set of models obtained using the method of [5]. This is due to the particular structural transformation used during the translation from modal formulae into first-order formulae. For instance, with the 3MG calculus the only minimal modal Herbrand model of the tableau clause $\Delta = w : (p_1 \vee [R_1]p_2)$ is the empty set. If we translate Δ into a first-order formula introducing a new symbol corresponding to $[R_1]p_2$, then in the resulting set of clauses there is a positive clause of the form $P_1(w) \vee Q_{[R_1]p_2}(w)$, which may produce a model not produced by the 3MG calculus. This implies it is possible to

create models that are not minimal, at least not under our definition of minimal modal Herbrand model, for the original modal formula, but are minimal for the first-order clauses.

Even though there is some similarity between some of the rules, the 3MG calculus and the PUHR calculus [3] do not correspond directly to each other in the sense that a step in the modal calculus can be simulated by one or more steps of the calculus of the PUHR approach, or the other way around. As said in Section 4, and explained in detail in [15], we prove minimal model soundness and completeness of the 3MG calculus by showing that the two calculi are approximations of each other via a non-trivial new translation of modal formulae to first-order clause form. Our new translation has the additional benefit that it leads to the reduction of the search space in the application of the PUHR approach to clauses obtained by the standard translation.

The use of the (*SBR*) rule in the 3MG calculus represents another important difference to the PUHR approach. It is possible to use a modal version of the PUHR rule where the box formulae are expanded away into disjunctions of tableau literals instead of the (*SBR*) rule. The benefits would be fewer inference steps and no intermediary clauses would be produced. It is not clear whether this would improve efficiency significantly though. A successful application of a multiple premise rule like the PUHR rule requires combinatorially many matching attempts to find the right premises for performing an inference step. Thus, we do not expect there to be significant degrading in performance through the use of the (*SBR*) rule. As the (*SBR*) rule can be viewed as the composition of the standard closure rule and expansion rules for multi-modal tableaux calculi, it has the advantage that it is more ‘modal’, and thus more natural, than a modal tableau version of the PUHR rule, which would require the introduction of more notation non-standard for modal tableau calculi.

In [7] the authors present a tableau calculus for circumscriptive reasoning for the description logic \mathcal{ALCO} . Circumscription aims to minimize a specific set of predicates. This means that our calculus can be thought as the particular circumscription case in which all predicates are minimized. However, the calculus presented in [7] differs from the 3MG calculus in aim, the logic considered, and the methodology used. First, the calculus in [7] does not actually generate minimal models (because *R* predicates cannot be minimised). Its task is to check if a formula is entailed by an interpretation respecting a circumscriptive pattern. Second, their tableau calculus is for the description logic \mathcal{ALCO} which has different expressive power compared to multi-modal logic $\mathbf{K}_{(m)}$ extended with reflexivity and symmetry. Finally, their method involves the use of intermediate tests during the tableau derivation, a technique also used in [12] that presents a tableau calculus for circumscription in function-free first-order clauses. The 3MG calculus does not require any separate test, because minimality is ensured by the (*CS*) rule and model constraint propagation.

The complement splitting rule (*CS*) in the 3MG calculus is based on the *neg* function rather than standard negation, because standard negation of diamond

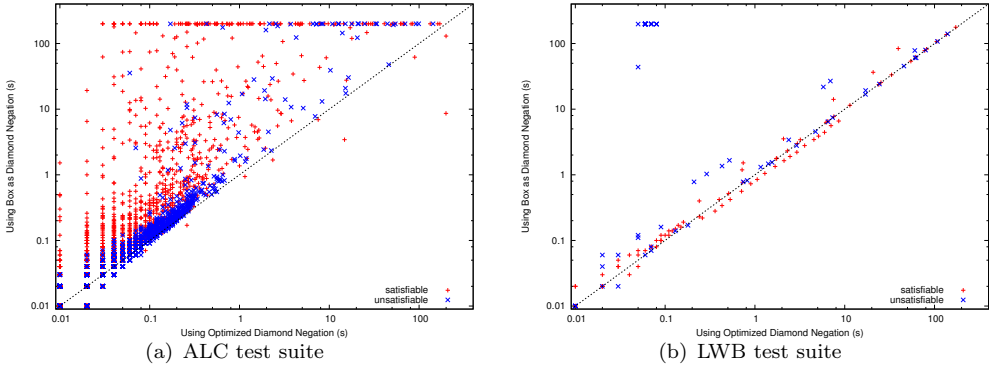


Fig. 3. Comparison of CPU runtimes in seconds of using the *neg* function

formulae can lead to non-minimal models. For example, given a positive clause $w : p_1 \vee w : \langle R_1 \rangle \neg p_2$, the negation of $w : \langle R_1 \rangle \neg p_2$ is $w : [R_1]p_2$. The expansion of the box formula could result in a non-minimal model. The problem is that a model created on this branch might include positive literals originating from below the box operator. This could cause a minimal model to be missed. For this reason the *neg* function defines the negation of a labelled diamond formula as the negation of the relation that this diamond would create.

With the view of using this technique for other related logics, it is important to note that it is not possible to use the *neg* function if the negated labelled relation resulting from it can be generated by rules other than the (\diamond) rule.

The use of a complement splitting rule in tableau calculi is not new, though typically the rule would be used in the form $u : \phi_1 \vee \phi_2 / u : \phi_1 \mid u : \phi_2, u : \neg \phi_1$. This ensures that branches are disjoint and avoids repeated inference steps, but can lead to significant degrading of performance due to the additional expansion of $u : \neg \phi_1$. To test whether our form of complement splitting based on the *neg* function can be an effective way of achieving a reduction in search space without degrading the performance, we have implemented a simple tableau prover for $\mathbf{K}_{(m)}$. The main optimization in the implementation is the use of backjumping, a clever form of backtracking. The different diamond negation is obtained by setting a flag. We decided to implement complement splitting only for diamond formulae to avoid inferences due to the negation of other complex formulae like box formulae or conjunctions. Tests were run with a timeout of 200 seconds on a Dell machine with Intel Core2 Duo Processor E6300 (2M Cache, 1.86 GHz), and 1GB of main memory. Figure 3 shows the CPU time comparisons with and without our optimized diamond negation over the ALC test suite from <http://www.csc.liv.ac.uk/~ullrich/ALC1996/> (collection of problems in alc-3-1-x-3-2 and alc-3-1-x-3-5), and the LWB test suite from <http://iamwww.unibe.ch/~lwb/benchmarks/benchmarks.html>. That there are more points above the diagonal of both graphs indicates superior performance for complement splitting with optimized diamond negation for both test suites. For the LWB test suite the advantage is however less prominent. This is perhaps not surprising because the problems in the ALC test suite are known to contain more

redundancies. It is interesting to note that almost all the cases in which the standard negation works better in the LWB test suite are for the class k_grz_n , while for all other classes the result is either in favour of our optimized negation or there is no evident difference.

6 Conclusion

We introduced the notion of (minimal) modal Herbrand model and defined a multi-modal tableau calculus which generates all and only minimal modal Herbrand models. We proved minimal model soundness and completeness by devising a new translation from modal formulae into first-order clause sets, and showing that there exists a correspondence between the 3MG calculus and a variation of the PUHR approach based on selection-based resolution. A positive side-effect of the function *neg* for the negation of diamond formulae is a reduction of the search space for the 3MG approach as the experiments have shown (we believe these can be carried over to the PUHR approach, as we intend to show in a future work).

A possible extension of our approach includes lazy clausification performed by appropriate extension rules. This would not improve the theoretical worst-case complexity of the calculus but will be of practical benefit in a future implementation. The calculus presented in this paper does not use lazy clausification because it makes it easier to prove its soundness and completeness via the connection with the PUHR approach, which is a purely clausal approach.

This work is just a starting point of developing more general tableau calculi for minimal model generation. Extensions in several directions are possible. In this paper we focused only on minimal model generation of multi-modal logic $\mathbf{K}_{(m)}$ and its extension with axioms **T** and **B**. An interesting extension is the addition of other well-known axioms such as **D**, **4** and **5** that correspond to seriality, transitivity and euclideaness of the accessibility relations. The main challenge is to understand how to deal with the introduction of a blocking technique, which would avoid the possibility of generating infinite models without compromising soundness and completeness of the calculus. The minimal model generation for dynamic modal logics [17] and description logics, such as *ALCO* and *ALCOIQ* (for which circumscription has been studied in [2,7]), is another possible direction for future work.

References

- [1] P. Baumgartner, P. Fröhlich, U. Furbach, and W. Nejdl. Tableaux for diagnosis applications. In *Proc. TABLEAUX'97*, volume 1227 of *LNCS*, pages 76–90. Springer, 1997.
- [2] P. A. Bonatti, C. Lutz, and F. Wolter. The complexity of circumscription in description logic. *J. Artif. Int. Res.*, 35:717–773, 2009.
- [3] F. Bry and A. Yahya. Positive unit hyperresolution tableaux and their application to minimal model generation. *J. Automated Reasoning*, 25(1):35–82, 2000.
- [4] H. De Nivelle, R. A. Schmidt, and U. Hustadt. Resolution-based methods for modal logics. *Logic J. IGPL*, 8(3):265–292, 2000.

- [5] L. Georgieva, U. Hustadt, and R. A. Schmidt. Computational space efficiency and minimal model generation for guarded formulae. In *Proc. LPAR'01*, volume 2250 of *LNAI*. Springer, 2001.
- [6] L. Giordano, V. Gliozzi, N. Olivetti, and G. L. Pozzato. Reasoning about typicality in preferential description logics. In *Proc. JELIA'08*, volume 5293 of *LNCS*, pages 192–205. Springer, 2008.
- [7] S. Grimm and P. Hitzler. A preferential tableaux calculus for circumscriptive \mathcal{ALCO} . In *Proc. RR'09*, volume 5837 of *LNCS*, pages 40–54. Springer, 2009.
- [8] J. Hintikka. Model minimization - an alternative to circumscription. *J. Automated Reasoning*, 4(1):1–13, 1988.
- [9] U. Hustadt and R. A. Schmidt. Using resolution for testing modal satisfiability and building models. *J. Automated Reasoning*, 28(2):205–232, 2002.
- [10] S. Lorenz. A tableaux prover for domain minimization. *J. Automated Reasoning*, 13(3):375–390, 1994.
- [11] J. McCarthy. Circumscription - a form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.
- [12] I. Niemelä. Implementing circumscription using a tableau method. In *Proc. ECAI '96*, pages 80–84. Wiley, 1996.
- [13] I. Niemelä. A tableau calculus for minimal model reasoning. In *Proc. TABLEAUX'96*, volume 1071 of *LNCS*, pages 278–294. Springer, 1996.
- [14] H. J. Ohlbach, A. Nonnengart, M. de Rijke, and D. M. Gabbay. Encoding two-valued nonclassical logics in classical logic. In J. A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, pages 1403–1486. Elsevier, 2001.
- [15] F. Papacchini and R. A. Schmidt. A tableau calculus for minimal modal model generation. Preprint, School of Computer Science, University of Manchester. <http://www.cs.man.ac.uk/~papacchf/PapacchiniSchmidt11a.pdf>.
- [16] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, 1987.
- [17] R. A. Schmidt and U. Hustadt. First-order resolution methods for modal logics, 2006. Manuscript. To appear in *Volume in memoriam of Harald Ganzinger*. <http://www.cs.man.ac.uk/~schmidt/publications/SchmidtHustadt06a.html>.