Computing Minimal Models Modulo Subset-Simulation for Modal Logics*

Fabio Papacchini and Renate A. Schmidt

The University of Manchester, UK
{papacchf,schmidt}@cs.man.ac.uk

Abstract. In this paper we propose a novel minimality criterion for models of modal logics based on a variation of the notion of simulation, called subset-simulation. We present a minimal model sound and complete tableau calculus for the generation of this new kind of minimal models for the multi-modal logic $\mathbf{K}_{(m)}$, and we discuss extensions to cover more expressive logics. The generation of minimal models is performed incrementally by using a minimality test to close branches representing non-minimal models, or to update the set of minimal models. Subset-simulation minimal models have the advantage that they are semantically more natural than models obtained by using syntactic minimality criteria.

1 Introduction

For fault analysis, verification of systems and validation of the logical formalisation of an application, model generation methods are useful for finding counterexamples as a means of debugging [14]. Models can be generated using tableau methods. For example, Smullyan-type labelled tableau calculi can be used to generate the essential parts of any model. However, even for the most well-behaved, decidable logics, in general, there are uncountably many different models for satisfiable formulae and models can be very large. The import of Herbrand's theorem is that we can restrict attention to the class of Herbrand models, because they are kinds of canonical models sufficient for showing soundness and completeness of many deductive systems. For the purposes of model generation, the class of Herbrand models has the advantage that it can be ordered by the subset relation. It is thus possible to focus on generating models minimal under this ordering. Generating minimal Herbrand models for classical logics has been studied in [4, 12] and for modal logics in [13]. For the modal logics **K**, **KT**, **KB**, **KTB** it has been shown minimal Herbrand models are finite [13], but for other extensions of **K** minimal Herbrand models are in general infinite.

By contrast, domain minimal models are finite for all logics with the finite model property. Another possibility therefore is to focus on the generation of models with minimised domains. Domain minimal models, however, tend to be counter-intuitive for verification and debugging purposes because too many

* The first author is supported by an EPSRC EU Doctoral Training Award

Table 1. Modalities and their corresponding frame conditions

$[R_i]$	Axiom	Frame condition
K		
T	$[R_i]p \rightarrow p$	reflexivity
B	$p \rightarrow [R_i]\langle R_i\rangle p$	symmetry
D	$[R_i]p \rightarrow \langle R_i\rangle p$	seriality
4	$[R_i]p \rightarrow [R_i][R_i]p$	transitivity
5	$\langle R_i\rangle p \rightarrow [R_i]\langle R_i\rangle p$	Euclideanness

worlds are collapsed to a single world. For instance, in a modal logic with doxastic modalities there are models in which belief states (those in the image of the belief relation) are identified and reflexive loops created. In most formalisations the belief relation is however not reflexive. This means there is a need to find classes of models better suited for debugging purposes.

As Herbrand models are too large and domain minimal models are too small, in this paper we study subset-simulation minimal models as a middle ground between the two. Subset-simulation is a relationship between models based on a variation of the notion of simulation [5, 7, 8]. Being applied directly on the graph representation of models means subset-simulation minimality preserves the semantics in minimal models, and is suitable for a large number of non-classical logics. It also results in more natural and intuitive minimal models than minimal Herbrand models and domain minimal models (Section 3).

We present a tableau calculus designed to generate subset-simulation minimal models for the multi-modal logic $\mathbf{K}_{(m)}$ in Section 5. The tableau is minimal model complete, but it is not minimal model sound. That is, it generates all minimal models, but also non-minimal models are generated. Section 6 shows how the calculus can be extended with a minimality test, called subset-simulation test, in order to generate only minimal models and achieve minimal model soundness. The resulting approach iteratively computes exactly the models minimal modulo subset-simulation by updating the set of minimal models as the derivation proceeds. Although the calculus we present is for the multi-modal logic $\mathbf{K}_{(m)}$, extensions to cover more expressive logics are easy to obtain. We conclude the paper with a discussion of possible extensions of the calculus and remarks on implementation (Section 7).

2 Preliminaries

We work with modal formulae of propositional multi-modal logic $\mathbf{K}_{(m)}$ possibly extended with universal modalities or a subset of the well-known axioms **T**, **B**, **D**, **4**, and **5**. Table 1 lists the axioms and their semantic meaning.

A *modal formula* is a formula of the form $\top$, $\bot$, p_i, $\neg\phi$, $\phi_1 \wedge \phi_2$, $\phi_1 \vee \phi_2$, $\langle R_i\rangle\phi$, $[R_i]\phi$, $[\mathcal{U}]\phi$, $\langle\mathcal{U}\rangle\phi$, where $\top$ and $\bot$ are two nullary logical operators for, respectively, true and false; p_i is a propositional symbol; $\neg$, $\wedge$, $\vee$, $\langle R_i\rangle$, $[R_i]$ are,

respectively, the logical operators negation, conjunction, disjunction, diamond and box; $[\mathcal{U}]$ and $\langle\mathcal{U}\rangle$ are universal modalities; and ϕ_1, ϕ_2, ϕ are modal formulae.

We adopt the standard semantics of modal formulae known as Kripke semantics. A *frame* for multi-modal logics is a tuple $(W, \mathcal{R})$, where W is a non-empty set of worlds and $\mathcal{R} = \{R_1, \ldots, R_n\}$ is a set of accessibility relations over W. An *interpretation* $\mathcal{I}$ is a tuple $(W, \mathcal{R}, V)$ composed of a frame and an interpretation function V that assigns to each world $u \in W$ a set propositional symbols meaning that such propositional symbols hold in u. Given an interpretation $\mathcal{I} = (W, \mathcal{R}, V)$ and a world $u \in W$, truth of a modal formula ϕ is inductively defined as follows.

$\mathcal{I}, u \not\models \bot$	$\mathcal{I}, u \models \top$
$\mathcal{I}, u \models p_i$	iff $p_i \in V(u)$
$\mathcal{I}, u \models \neg\phi$	iff $\mathcal{I}, u \not\models \phi$
$\mathcal{I}, u \models \phi_1 \vee \phi_2$	iff $\mathcal{I}, u \models \phi_1$ or $\mathcal{I}, u \models \phi_2$
$\mathcal{I}, u \models \phi_1 \wedge \phi_2$	iff $\mathcal{I}, u \models \phi_1$ and $\mathcal{I}, u \models \phi_2$
$\mathcal{I}, u \models [R_i]\phi$	iff for every $v \in W$ if $(u, v) \in R_i$ then $\mathcal{I}, v \models \phi$
$\mathcal{I}, u \models \langle R_i\rangle\phi$	iff there is a $v \in W$ such that $(u, v) \in R_i$ and $\mathcal{I}, v \models \phi$
$\mathcal{I}, u \models [\mathcal{U}]\phi$	iff for every $v \in W$ $\mathcal{I}, v \models \phi$
$\mathcal{I}, u \models \langle\mathcal{U}\rangle\phi$	iff there is a $v \in W$ such that $\mathcal{I}, v \models \phi$

Given an interpretation $\mathcal{I}$, a world u and a modal formula ϕ, if $\mathcal{I}, u \models \phi$ holds, then $\mathcal{I}$ is a *model* of ϕ.

3 Subset-Simulation as Minimality Criterion

Subset-simulation is a variation of the notion of simulation [8, 7, 5], and is known from [1, 11], where it is used for the description logic $\mathcal{EL}$ and is simply called simulation. As we use both simulation and its variation, we decided to call the latter subset-simulation.

Let $M = (W, \mathcal{R}, V)$ and $M' = (W', \mathcal{R}', V')$ be two models of a modal formula ϕ. A *simulation* is a binary relation $S \subseteq W \times W'$ such that for any two worlds $u \in W$ and $u' \in W'$, if uSu' then the following hold.

- $V(u) = V'(u')$ and
- if uRv for some $R \in \mathcal{R}$, then there exists a $v' \in W'$ such that $R \in \mathcal{R}'$, $u'Rv'$, and vSv'.

Let $M = (W, \mathcal{R}, V)$ and $M' = (W', \mathcal{R}', V')$ be two models of a modal formula ϕ. A *subset-simulation* is a binary relation $S_\subseteq \subseteq W \times W'$ such that for any two worlds $u \in W$ and $u' \in W'$, if $uS_\subseteq u'$ then the following hold.

- $V(u) \subseteq V'(u')$ and
- if uRv for some $R \in \mathcal{R}$, then there exists a $v' \in W'$ such that $R \in \mathcal{R}'$, $u'Rv'$ and $vS_\subseteq v'$.

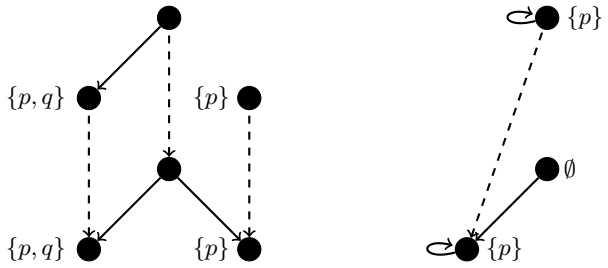

Fig. 1. Simulations between symmetric models modulo subset-simulation

If S is such that for all $u \in W$ there is at least one $u' \in W'$ such that uSu', then we call S a *full (subset-)simulation* from M to M'. We say a (subset-)simulation S is a *maximal (subset-)simulation* if there is no other (subset-)simulation $S' \neq S$ such that $S \subset S'$. Given two models M and M', if there is a full (subset-)simulation S from M to M', we say that M' *(subset-)simulates* M, or M *is (subset-)simulated* by M'.

In this paper we are only interested in full and maximal (subset-)simulations. For this reason, when we refer to (subset-)simulations we mean full and maximal (subset-)simulations. Where ambiguous, we explicitly state what kind of (subset-)simulation we mean.

Subset-simulation has properties that allow us to use it to define a minimality criterion for modal logic models. Specifically, subset-simulation is a reflexive and transitive relation on models. Hence, it forms a preorder on models. This means that we can consider as minimal models all the models that are minimal with respect to subset-simulation. As subset-simulation is not anti-symmetric (which means it is not a partial order), it is possible for models to form a symmetry class. A *symmetry class* is a set of models that subset-simulate each other. This may result in minimal models belonging to large symmetry classes, and therefore also a large number of minimal models. To overcome this problem, we aim to be more restrictive by using also the notion of simulation within a symmetry class of minimal models.

Let M be a model of a modal formula ϕ. M is *minimal modulo subset-simulation* iff for any other model M' of ϕ, if M subset-simulates M', then M' subset-simulates M and either there is no simulation relationship between M and M', or M' simulates M.

The use of a simulation check within a symmetry class allows us to recognise bisimilar models, models that are embedded in other models, and to impose an extra ordering over symmetric models. Figure 1 shows two examples of what kinds of minimal models are excluded due to the use of simulation. The two models on the right belong to a symmetry class, and the two models on the left belong to a different symmetry class (that is, the models on the right subset-simulate each other, and the models on the left subset-simulate each other). The dashed lines in the figure represent the simulation relationships between

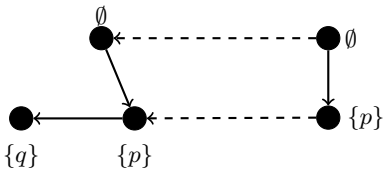

Fig. 2. Minimality modulo subset-simulation vs. minimal Herbrand models

the models. As the models at the bottom simulates the models at the top, only the two models at the top are minimal modulo subset-simulation.

Models minimal modulo subset-simulation have interesting properties. First, the interpretation function is minimal with respect to a given frame. This means that given a set of worlds and the accessibility relations between them, the interpretation function assigns to each world the minimal number of propositional variables such that the resulting interpretation is a model for a given formula. Second, as subset-simulation is directly based on the graph structure of models, the minimality criterion is able to discern models semantically, thus avoiding semantically redundant minimal models (in opposition to the minimal Herbrand models criterion). In other words, the criterion is able to compare models having distinct domains by comparing directly the labelling functions and the accessibility relations. Being based on the graph structure of models makes minimality modulo subset-simulation a criterion suitable for a large number of modal logics. Finally, subset-simulation gives priority to finite loop-free models, meaning that usually models minimal modulo subset-simulation are not domain minimal.

We conclude this section with two examples of models minimal modulo subset-simulation in order to compare the notion with other minimality criteria. The first example shows that the new minimality criterion does not suffer the syntactic restriction that affects Herbrand models. For lack of space we cannot give the formal definition of modal Herbrand models for which we refer to [13]. Even though there are a few differences, it might help to think of them as the Herbrand models of the translation of a modal formula into a first-order formula. Let us consider the modal formula $\phi = \langle R_1 \rangle p \vee \langle R_1 \rangle (p \wedge \langle R_1 \rangle q)$. The minimal Herbrand models of ϕ are shown in Figure 2. As can be seen, the model on the right is completely embedded in the model on the left. Due to the syntactic restrictions of Herbrand models, however, it is not possible to recognise this relation between models, and the method proposed in [13] would consider both the as being minimal models. By contrast, subset-simulation minimality considers only the model on the right as minimal because it is subset-simulated by the other model, but not the other way around.

The second example shows that models minimal modulo subset-simulation are more natural than domain minimal models. Consider the formula $\phi = \langle has\_father \rangle doctor$. Figure 3 shows two models that satisfy ϕ. The left model in the figure is the domain minimal model, and the right model is the model minimal modulo subset-simulation. In the domain minimal model ϕ is satisfied

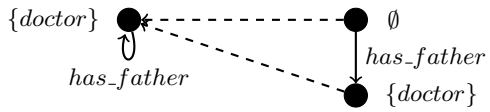

Fig. 3. Minimality modulo subset-simulation vs. domain minimality

by creating a loop, meaning that there is a person who is their own father and who is a doctor. Even though such a model satisfies ϕ, it does not reflect our intuition of the has_father relation. This problem is avoided in the model minimal modulo subset-simulation, where a new successor is required, thus ensuring that a person is not their own father. Admittedly this is a simple example, but it illustrates problems avoided for relations where reflexivity is counter-intuitive. Our main point is avoiding loops in models if they are not necessary for the finiteness of the model, and we only create loops containing the least positive information by minimising the interpretation function when necessary.

4 Computing Subset-Simulation between Models

Even though not concerned with modal logic models, the paper [8] presents algorithms for computing simulations between graphs. One of the algorithms presented in [8] computes maximal self-simulation, which is the maximal simulation between a graph and itself. This algorithm can be modified for computing full and maximal subset-simulation between two models of a modal formula ϕ.

Figure 4 shows the pseudo-code of the algorithm that takes as input two models $M = (W, \mathcal{R}, V)$ and $M' = (W', \mathcal{R}', V')$, and returns the full and maximal subset-simulation from M to M' or the empty set, if there is no full subset-simulation. The following variables and functions are used in the algorithm.

- $outrel(u)$ returns the set of outgoing accessibility relations from u.
- $sim(u)$ represents the set of worlds in W' that are subset-similar to u.
- $post(R_i, u)$ returns all the R_i-successors of u.
- $pre(R_i, u)$ returns all the R_i-predecessors of u.
- $pre(R_i, W)$ is the union of the sets resulting by the application of the pre function with respect to the relation R_i to all the elements of W, that is, $pre(R_i, W) = \bigcup_{u \in W} pre(R_i, u)$.

The basic idea behind the algorithm is that subset-simulation is computed by overestimating the possible subset-simulation, and then pruning false guesses by checking the second property of subset-simulation. To obtain the algorithm in Figure 4 from the algorithm presented in [8], the following must be taken into consideration. First, if a full subset-simulation does not exist, then the empty set is returned. Second, the input is composed of two different models, this is because we are not interested in self subset-simulations. Reflexive edges need to be correctly handled. The two graphs have more than one accessibility relation. This last point is the reason why the algorithm loops over the set of accessibility

```
1: for all v ∈ W do
2:     sim(v) ← {u ∈ W' | V(v) ⊆ V'(u) and outrel(v) ⊆ outrel(u)}
3:     if sim(v) = ∅ then
4:         return ∅
5: for all R_i ∈ R do
6:     for all v ∈ W do
7:         remove(v) ← pre(R_i, W') \ pre(R_i, sim(v))
8:     while there is a vertex v ∈ W s.t. remove(v) ≠ ∅ do
9:         aux_remove_v ← ∅
10:        for all u ∈ pre(R_i, v) do
11:            for all w ∈ remove(v) do
12:                if w ∈ sim(u) then
13:                    sim(u) ← sim(u) \ {w}
14:                    for all w' ∈ pre(R_i, w) do
15:                        if post(R_i, w') ∩ sim(u) = ∅ then
16:                            if u = v then
17:                                aux_remove_v ← aux_remove_v ∪ {w'}
18:                            else
19:                                remove(u) ← remove(u) ∪ {w'}
20:            if sim(u) = ∅ then
21:                return ∅
22:        remove(v) ← aux_remove_v
23: return {(u, v) | u ∈ W and v ∈ sim(u)}
```

Fig. 4. Pseudo-code for computing full, maximal subset-simulation between two models

relations and refines the subset-simulation by incrementally computing the intersection of the subset-simulation with respect to a single accessibility relation. For reason of space we omit a more detailed description of the algorithm.

Soundness of computing the subset-simulation in this way is a consequence of the following theorem.

Theorem 1. *Let $M = (W, \mathcal{R}, V)$ and $M' = (W', \mathcal{R}', V')$ be two models of a modal formula ϕ such that M is subset-simulated by M'. The full and maximal subset simulation $S_\subseteq$ can be computed as the intersection of all the full and maximal subset-simulations with respect to each single accessibility relation $R \in \mathcal{R}$.*

5 Tableau Calculus

We present a generic tableau calculus for the generation of minimal models modulo subset-simulation for the multi-modal logic $\mathbf{K}_{(m)}$. The calculus is generic in the sense that it can be easily extended to more expressive modal logics. Such extensions are discussed in Section 7.

The input of the calculus is a modal formula in negation normal form labelled by an initial world u. Transformation to negation normal form is not essential,

Table 2. Rules of the basic tableau calculus

$$(\alpha)\ \frac{u : (\phi_1 \wedge \ldots \wedge \phi_n) \vee \Phi^+_\alpha}{\begin{array}{c} u : \phi_1 \vee \Phi^+_\alpha \\ \vdots \\ u : \phi_n \vee \Phi^+_\alpha \end{array}} \qquad (\Box)\ \frac{(u,v) : R_i \quad u : [R_i]\phi}{v : \phi}$$

$$(\beta)\ \frac{u : \mathcal{A} \vee \Phi^+}{\begin{array}{l|l} u : \mathcal{A} & u : \Phi^+ \\ u : neg(\Phi^+) & \end{array}}$$

where $\mathcal{A}$ is of the form $\langle R_i\rangle\phi$, $[R_i]\phi$, or p_i, and $neg(\Phi^+) = \neg p_1 \wedge \ldots \wedge \neg p_n$, where each p_i is a disjunct of Φ^+.

$$(\Diamond)\ \frac{u : \langle R_i\rangle\phi}{\begin{array}{l|l|l|l} (u,u_1) : R_i & \ldots & (u,u_n) : R_i & (u,v) : R_i \\ u_1 : \phi & & u_n : \phi & v : \phi \end{array}}$$

where each u_i appears on the branch, and v is fresh.

$$(SBR)\ \frac{u : p_1, \ldots, u : p_n \quad u : \neg p_1 \vee \ldots \vee \neg p_n \vee \Phi^+_\alpha}{u : \Phi^+_\alpha}$$

but it simplifies the presentation. It also means that there is no need for preprocessing before applying the calculus, and it allows us to reduce the number of rules in the calculus.

In the calculus, disjunctions and conjunctions are assumed to be flattened for example, we write $\phi_1 \vee \phi_2 \vee \phi_3$ instead of $\phi_1 \vee (\phi_2 \vee \phi_3)$. By $\mathcal{A}$ we mean a modal formula of the form p_i, $\langle R_i\rangle\phi$ or $[R_i]\phi$. We use Φ^+ to denote a non-empty disjunction, where all disjuncts are of the form $\mathcal{A}$, and Φ^+_α to denote a possibly empty disjunction where all disjuncts are of the form $\mathcal{A}$ or are conjunctions. By $neg(\Phi^+)$ we mean the conjunction $\neg p_1 \wedge \ldots \wedge \neg p_n$, where the p_i are all the positive propositional variables appearing as disjuncts of Φ^+. If Φ^+ does not contain any p_i, then $neg(\Phi^+) = \top$.

Table 2 presents the rules of the calculus for the multi-modal logic $\mathbf{K}_{(m)}$. A branch $\mathcal{B}$ of the tableau is a sequence $N_0, N_1, \ldots, N_i$ of sets of formulae of the form $u : \phi$ or $(u,v) : R$, where $N_0 = \{u : \phi\}$ and ϕ is the input formula. Given an input formula $u : \phi$, the rules of the calculus are exhaustively applied. At most one rule is applied to any formula appearing as the main premise, where the main premise of multi-premises rules is the premise on the right. For fairness, each instance of a rule application is applied exactly once. Each rule application extends the current branch. That is, a rule applied to a formula in the set N_i extends the branch with the set N_{i+1}, where N_{i+1} is the set N_i plus the conclusions of the applied rule. Given an open branch $\mathcal{B}$, a model $M = (W, \mathcal{R}, V)$ can be extracted from $\mathcal{B}$ as follows. The domain W is the set of all the labels in $\mathcal{B}$, the accessibility relations are composed of all the instances $(u,v) \in R_i$ in $\mathcal{B}$, the interpretation function V is such that $V(u) = \{p_i \mid u : p_i \in \mathcal{B}\}$.

Even though the input formula is in negation normal form, the calculus can be thought of as a calculus for formulae in clausal form. This is achieved by the (α) rule that not only deals with conjunctions, but also performs lazy clausification. If such a lazy clausification is performed in a clever way, for example, by using a good heuristic for choosing the right conjunction to expand, it can result in the reduction of inferences due to the implicit restriction of Φ^+_α in the premise of the rule. For instance, let us assume that the premise is $u : (p_1 \wedge p_2) \vee (\neg p_3 \wedge \neg p_4)$. If the (α) rule is applied to the first conjunction, it results in the two modal formulae $u : p_1 \vee (\neg p_3 \wedge \neg p_4)$ and $u : p_2 \vee (\neg p_3 \wedge \neg p_4)$. The (α) rule is again applicable to both of them. If instead, the (α) rule is applied to the other conjunction first, the resulting formulae are $u : \neg p_3 \vee (p_1 \wedge p_2)$ and $u : \neg p_4 \vee (p_1 \wedge p_2)$, and the (α) rule is not applicable to any of them.

The $(\Box)$ rule is the common rule for box formulae, and simply expands formulae in the scope of a box modality as required by the semantics of box formulae.

The (β) rule is one of the two branching rules of the calculus. Its purpose is to branch over disjunctions without any negated propositional variables, and to close the left branch if it is not minimal. This latter point is achieved by the use of a limited form of complement splitting (a more common use of complement splitting can be found in [4]). The reason why complement splitting is applied only on positive propositional variables is that the negation of diamond formulae or box formulae would result in new modal formulae (specifically, box formulae and diamond formulae) that can compromise the minimality of the resulting model. For example, let us assume that the (β) rule is applied to $u : p \vee [R_1]q$. If the complement $\langle R_1 \rangle \neg q$ of $[R_1]q$ would have been added to the left branch, the left branch would still be open, and the resulting model would still be a model for the original formula, but the newly introduced diamond formula would generate unnecessary information. The resulting model would not be minimal. A similar example can be given for the case of the negation of diamond formulae.

The $(\Diamond)$ rule is the expansion rule for diamond formulae. As it can lead to the expansion of a diamond formula in all possible worlds plus a fresh one, it is an expensive rule. It is, however, required to achieve minimal model completeness. This rule is known from literature, for example [9, 10, 3]. It is worth pointing out that the $(\Diamond)$ rule in general does not guarantee termination for the purpose of minimal model generation, but it ensures termination in case we are only interested in checking the satisfiability of a modal formula belonging to a logic with the finite model property. The termination issue for minimal model generation does not affect the multi-modal logic $\mathbf{K}_{(m)}$, but it has to be taken into consideration when generalising to more expressive logics.

Finally, the (SBR) rule is a selection-based resolution rule. It can be seen as a weaker version of the (SBR) rule in [13], the $PUHR$ rule in [4], or the hypertableau rule in [2]. The aim of this rule is twofold. First, it provides the closure rule of the calculus, because atomic closure is sufficient. Second, it allows to remove negative information (that is, all negative propositional variables) from a disjunction. The reason behind the (SBR) rule is that if a disjunction contains negative information (that is, at least one negated propositional variable) that

is not in conflict with any formula on the branch, then any expansion of such a disjunction results in either the minimal model, where the disjunction is true due to the negative information, or in a non-minimal model. Hence, there is no advantage in expanding a disjunction as long as it is not possible to remove all the negative information from it. The (SBR) rule is the reason why other rules, specifically the (β) rule and the (α) rule, can be applied only to disjunctions of the form Φ^+ or Φ^+_α. This decreases the number of required inferences.

The calculus presented so far does not yet constitute the full method for the generation of models minimal modulo subset-simulation, but is a starting point for it.

Theorem 2. *The tableau calculus is refutationally sound and complete for* $\mathbf{K}_{(m)}$.

For lack of space we do not provide a formal proof, but the calculus does not differ much from known calculi. All the rules have already been applied in other calculi, or are sound variations of common rules. The main differences are due to variations of rules in order to not expand formulae that are already minimally satisfied, for example, the restrictions due to Φ^+ or Φ^+_α.

Theorem 3. *The tableau calculus is subset-simulation minimal model complete. That is, it generates all models minimal modulo subset-simulation.*

Proof. Suppose $M, u \models \phi$, where $M = (W, \mathcal{R}, V)$ is a model minimal modulo subset-simulation, $u \in W$ and ϕ is a modal formula. We first show that the tableau having as input $u : \phi$ has an open, fully expanded branch $\mathcal{B} = N_0, \ldots, N_i, \ldots$, where $N_0 = \{u : \phi\}$ and for all $i \geq 0$ the following holds: $M \models N_i$ implies $M \models N_{i+1}$, where $M \models N_i$ means that for each formula $u : \phi \in N_i$ we have that $M, h(u) \models \phi$, where h is a function mapping labels in N_i to domain elements in M such that $h(u) = u$ for the starting label u. Suppose N_{i+1} is obtained from N_i by the application of a rule ρ. We consider several cases.

ρ is the $(\Box)$ rule. This means the expanded formula is a labelled box formula $u : [R_i]\phi'$, and $(u, v) : R_i$ is in N_i for some v. As $M \models N_i$, we have that $M, h(u) \models [R_i]\phi'$ and $(h(u), h(v)) \in R_i$. This implies $M, h(v) \models \phi'$. That is, M is a model for the conclusion of the application of the $(\Box)$ rule.

ρ is the (α) rule. This means that the expanded formula is a labelled disjunction, where at least one disjunct ϕ_α is a conjunction. As $M \models N_i$, we have that $M, h(u) \models \phi$. This implies that $M, h(u) \models \phi'_\alpha$, where ϕ'_α is the result of distributing the conjunction of ϕ_α over ϕ. Hence, M is a model for all the conjuncts of ϕ'_α. That is, M is a model for the conclusions of the application of the (α) rule.

ρ is the $(\Diamond)$ rule. This means that the expanded formula is a labelled diamond formula, let us say $u : \langle R_i\rangle\phi'$. As $M \models N_i$, we have that $M, h(u) \models \langle R_i\rangle\phi'$. This implies that there exists an R_i-successor v of $h(u)$ such that $M, v \models \phi'$. If there is no w in N_i such that $h(w) = v$, then we choose the right-most conclusion of the $(\Diamond)$ rule and let $h(w) = v$. If there is already a world w in N_i such that $h(w) = v$, then choose the conclusion where w is used as witness.

ρ is the (β) rule. This means that the expanded formula is a labelled disjunction where the disjuncts are propositional variables, diamond formulae or box formulae. As $M \models N_i$, we have that $M, h(u) \models \phi$. This implies that M satisfies at least one of the disjuncts. Suppose $\mathcal{A}$ is one such disjunct and Φ^+ is the remaining part of the disjunction. Assume $\mathcal{A}$ is expanded in the left branch, then two cases are possible. First, $M, h(u) \models neg(\Phi^+)$, that is, M is a model for all the conclusions in the left branch. Second, $M, h(u) \not\models neg(\Phi^+)$. That is, there is a propositional variable p_i that appears as disjunct in Φ^+ such that $M, h(u) \models p_i$. This means that $u : \phi$ is already satisfied because p_i is satisfied, that is, one of the disjunct of Φ^+ is satisfied. Hence, $M, h(u) \models \Phi^+$ and the correct expansion of N_i is the right branch of the (β) rule.

ρ is the (SBR) rule. This means that the expanded formula $u : \phi$ is of the form $u : \neg p_1 \vee \ldots \vee \neg p_n \vee \phi'$ and that $u : p_1, \ldots, u : p_n$ appear in N_i. As $M \models N_i$, we have that $M, h(u) \models \phi$ and $M, h(u) \models p_i$ for all i within $1 \leq i \leq n$. This implies that $M, h(u) \models \phi$ iff $M, h(u) \models \phi'$. That is, M is a model for the conclusion of the (SBR) rule.

This proves by induction that there is a branch $\mathcal{B}$ validated by M.

It remains to show that the model $M' = (W', \mathcal{R}', V')$ extracted from $\mathcal{B}$ is equivalent to M. From the construction of the branch, the domain of M' is such that $W'_h \subseteq W$, where $W'_h = \{v \mid h(u) = v \text{ for all } u \in W'\}$. This is, because the starting node u belongs to both W' and W, only applications of the $(\Diamond)$ rule create worlds, and these are mapped by following what holds in the minimal model M. The same reasoning is also applicable for the set of accessibility relations.

The interpretation function V' is such that for all $u : p_i \in \mathcal{B}$, $p_i \in V'(u)$. This implies that for all $u \in W'$ we have that $V'(u) \subseteq V(h(u))$. This is because $M \models N_i$ for all i and, specifically, $M, h(u) \models p_i$ for all $u : p_i \in \mathcal{B}$.

From these observations it follows that M' is either smaller (containing fewer worlds, fewer relational links, or there is some world for which the interpretation function is a subset of the interpretation function of M) or equal to M.

Assume that the frame of M' is smaller than the frame of M'. This implies M is not minimal because either M subset-simulates M' (when for some $u \in W'$ we have that $V'(u) \subset V(h(u))$) or M simulates M' (when for all $u \in W'$ we have that $V'(u) = V(h(u))$). The (subset-)simulation is simply the set $\{(u, h(u)) \mid u \in W'\}$. This contradicts the minimality of M. Hence, M' and M are based on the same frame.

Assume that for some $u \in W'$ we have that $V'(u) \subset V(h(u))$. This contradicts the assumption that M is a model minimal modulo subset-simulation because M' is subset-simulated by M (the subset-simulation is as in the previous case). This implies that for all $u \in W'$, $V'(u) = V(h(u))$.

As the frames and the interpretation functions of M and M' are the same, M and M' are the same model. This completes the proof. $\square$

6 Minimal Model Soundness

Although the calculus is minimal model complete as presented up to now, it is not yet minimal model sound. This means, although among all the generated models there are all the minimal ones, the calculus does not generate only minimal models. However, as the calculus is minimal model complete, minimal model soundness can be achieved by closing all the branches of the tableau from which non-minimal models can be extracted. In order to prune properly the search space, we introduce a minimality test called *subset-simulation test*. This test allows us either to detect non-minimal models before they are completely computed, or to refine the minimal models found so far (that is, updating the set of minimal models by deleting models and inserting a new one).

Following an idea in [4, 13], the aim of the minimal model test is to use previously extracted models to judge the minimality of the partial model that can be extracted from the currently selected branch. A crucial difference with [4, 13] is that we cannot guarantee that the first extracted model is minimal. Our solution is to compute minimal models incrementally, meaning it is only known at the end of the complete derivation whether a model is minimal. The incremental generation of minimal models is achieved in the calculus by always selecting the left-most branch with the least number of worlds for further expansion first. This means, the calculus generates first all the models with the smallest domain, and then incrementally increases the domain size of the generated models. This expansion strategy alone is not enough to make the calculus minimal model sound because domain minimal models have good chances of not being minimal. Nevertheless, we think this is a good heuristic, because the minimality test can only be performed by comparing already extracted models with one (partial) model, and because the complexity of the algorithm presented in Section 4 depends on the number of worlds in the two models. Hence, finding domain minimal models first is likely to speed up the incremental generation of minimal models.

The *subset-simulation test* is divided into two cases. First, M is the partial model extracted from an open but not fully expanded branch. If there exists an already extracted model M' that is subset-simulated by M and M is not subset-simulated by M', then M is not minimal and the branch from which it was extracted is closed.

Second, M is the model extracted from an open and fully expanded branch. Then M is compared with the already extracted models and branches are closed accordingly. The closure of branches involves consideration of these three cases.

- M subset-simulates some minimal model M', but M' does not subset-simulate M. This means M is not minimal, and the branch from which M was extracted must be closed.
- M does not subset-simulate any minimal model M', but M' subset-simulates M. This means M' and all the models belonging to the symmetry class of M' are not minimal, and the branches from which those models were extracted must be closed.
- Some minimal model M' subset-simulates M, and M subset-simulates M'. This means M belongs to the same symmetry class of M'. Hence, simulation

relationships between M and the models of the symmetry class need to be checked in order to refine the symmetry class. All the branches from models of the symmetry class which are no longer minimal must be closed.

The first case of the subset-simulation test allows us to prune the tree derivation before a branch is fully expanded. This is possible because if a partial model is already non-minimal, none of its possible extensions can be minimal. Hence, the branch can be closed without compromising minimal model completeness of the calculus.

As it is not always possible to recognise a non-minimal model before the branch is fully expanded, and because minimal models are computed incrementally by continuously refining the set of minimal models, the first case of the subset-simulation test is clearly not enough. The second case performs the refinement step of the current set of minimal models, meaning that even previously open and fully expanded branches can be closed. In other words, the second case requires checking subset-simulation relationships between M and representative models of all the symmetry classes of minimal models. This is because if M is subset-simulated by one model of a symmetry class, then it is subset-simulated by all of them due to subset-simulation being transitive.

From a theoretical perspective, it is not important when the minimality test is applied as long as it is always applied to open and full expanded branches (that is, as long as the second case of the minimality test is extensively performed). In order to avoid complex subset-simulation tests and to prune the derivation tree as soon as possible, heuristics can be used to fix the order of application of the rules and when the minimality test is performed. Our suggestion is to apply the rules in the following order: (SBR) rule, (α) rule, $(\Box)$ rule, (β) rule, and $(\Diamond)$ rule. The idea behind this order is to close a branch as soon as a contradiction occurs on the branch, and to delay the application of branching rules. Given this order of rule application, a sensible heuristic for the application of the minimality test is to perform it just before the application of the $(\Diamond)$ rule. This is because the $(\Diamond)$ rule has the highest branching factor, and the complexity of the subset-simulation test gradually increases after each application of this rule.

Using the proposed branch selection strategy and the minimality test, the calculus in Table 2 becomes minimal model sound.

Theorem 4. *Augmenting the tableau calculus with the subset-simulation test provides an approach that is minimal model sound when a fair expansion strategy is used. That is, it generates only models minimal modulo subset-simulation.*

7 Discussion

In the minimal model soundness theorem we required the expansion strategy of the calculus to be fair. The proposed expansion strategy to select the left-most branch with the least number of worlds can be seen as a variation of the common depth-first iterative deepening expansion strategy, where the weight used to select a branch is the number of worlds appearing on the branch. This

strategy is not the only possible fair expansion strategy that can be applied to the calculus. Other variations of the depth-first iterative deepening strategy or a breadth-first strategy can also be used, and the resulting procedure is still minimal model sound and complete. Among the common strategies, depth-first expansion is probably the only one that cannot be applied. This is because it is possible to have infinitely long branches, and depth-first expansion would not result in a complete tree derivation. As we have already pointed out, this is not the case for the multi-modal logic $\mathbf{K}_{(m)}$.

Even though the idea for the subset-simulation test is inspired by the model constraint propagation rule in [13, 4], there are differences to that minimality test. The main difference is that we need the complete tree derivation for establishing which models are minimal, while in [13, 4] this is not the case; the reason being that [13, 4] are concerned with the generation of minimal Herbrand models, which means a subset (the set of minimal models) of a subset (the set of Herbrand models) of all possible models. Minimality modulo subset-simulation, instead, needs to evaluate many more models. It is interesting to note that if minimality modulo subset-simulation is applied only to Herbrand models, then the resulting set of minimal models is a refinement of the set of minimal Herbrand models.

The calculus in Table 2 can be extended easily to cover extensions of modal logic $\mathbf{K}_{(m)}$ by introducing rules that properly deal with such extensions. Table 7 shows the rules that allow the expansion of the tableau calculus to modal logics enriched with universal modalities, or to extensions in which the accessibility relations satisfy frame conditions from Table 1. Any extended version of the calculus results in a minimal model sound and complete tableau calculus as long as the minimality test and the described expansion strategy are used. A property that could be lost is termination of the calculus. We have already pointed out that the $(\Diamond)$ rule does not guarantee termination for the purpose of subset-simulation minimal model generation.

The extensions allowed by the rules in Table 7 are not the only possible extensions. One of the advantages of using minimality modulo subset-simulation is that the minimality criterion is applied to the graph representation of models. This means that the minimality criterion can be applied to all non-classical logics defined by a Kripke semantics. This includes logics such as modal logics, description logics, and temporal logics (even those that are not translatable to fragments of first-order logic). It is known from the literature, for example [6], that bisimulation needs to be extended depending on the expressivity of the logic. This is because bisimulation, like simulation and subset-simulation, is a local definition. It is however not required to extend the notion of subset-simulation for minimality modulo subset-simulation because the criterion requires full subset-simulation, changing the scope of the definition from local to global.

From the point of view of implementation, the calculus presents several challenges. Many well-known optimisation techniques such as backjumping or unit propagation, or a variation of them can be applied to speed up the implementation. The main problem, depending on the logic under consideration, is the

Table 3. Structural rules for extending the calculus. Note: all worlds in the conclusion of a rule with empty premises must already appear on the branch

$$(\mathbf{T})\ \frac{}{(u,u):R_i} \qquad (\mathbf{B})\ \frac{(u,v):R_i}{(v,u):R_i}$$

$$(\mathbf{4})\ \frac{(u,v):R_i \quad (v,w):R_i}{(u,w):R_i} \qquad (\mathbf{5})\ \frac{(u,v):R_i \quad (u,w):R_i}{(v,w):R_i}$$

$$(\mathbf{D})\ \frac{}{(u,u_1):R_i \,|\, \ldots \,|\, (u,u_n):R_i \,|\, (u,v):R_i}$$

where u does not have an R_i-successor, each u_i appears on the branch, and v is fresh.

$$(\langle\mathcal{U}\rangle)\ \frac{u:\langle\mathcal{U}\rangle\phi}{u_1:\phi \,|\, \ldots \,|\, u_n:\phi \,|\, v:\phi}$$

where each u_i appears on the branch, and v is fresh.

$$([\mathcal{U}])\ \frac{u:[\mathcal{U}]\phi}{v:\phi}$$

where v appears on the branch.

possibility that the computation does not terminate. In this case, it might be sensible to impose a termination strategy at the cost of losing minimal model soundness and completeness, but preserving at least refutational soundness and completeness. This means not to stop the computation before the first model is found. After the first model has been found an early termination strategy can be used. This produces the best minimal models computed so far. As we are able to establish minimality of a model only in the complete derivation tree, stopping the computation at an early point does not guarantee the minimality of the models obtained so far. The idea of stopping the computation at a certain point can be seen as a branch and bound strategy, that is, the returned minimal models are the best minimal models extracted from the tableau up to this point. When to stop the derivation requires a new heuristic in the implementation, which one would probably make dependant on the domain of application. An alternative might be the use of a blocking mechanism such that the resulting procedure is strongly terminating. We are currently investigating blocking techniques to achieve strong termination while preserving minimal model soundness and completeness for logics with the finite model property. An appropriate blocking technique or a simplification of the $(\Diamond)$ rule might result in a more efficient tableau calculus.

8 Conclusion

We presented minimality modulo subset-simulation as a novel minimality criterion for modal logics. The minimal models obtained following this new minimal-

ity criterion have the benefit that they reflect the semantics of a modal formula in a more faithful way than other minimality criteria. Although we emphasised the application of the criterion to the multi-modal logic $\mathbf{K}_{(m)}$, its semantic nature makes it applicable to a large number of non-classical logics.

We presented a minimal model complete tableau calculus for the multi-modal logic $\mathbf{K}_{(m)}$, and discussed how to achieve minimal model soundness through the use of the subset-simulation test. The resulting minimal model sound and complete calculus can easily be expanded to cover extensions of the multi-modal logic $\mathbf{K}_{(m)}$.

Even though the expansion rule for diamond formulae is expensive and termination is not always guaranteed, we believe that variations of the calculus can be efficiently implemented in such a way that the generated models are semantically meaningful and useful for applications. An implementation of the calculus, its extensions and variations can give important further insight regarding the generation of minimal models for non-classical logics.

References

1. Baader, F.: Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles. In: IJCAI'03. pp. 319–324. Morgan Kaufmann (2003)
2. Baumgartner, P., Fürbach, U., Niemelä, I.: Hyper tableaux. In: JELIA'96. LNCS, vol. 1126, pp. 1–17. Springer (1996)
3. Bry, F., Torge, S.: A deduction method complete for refutation and finite satisfiability. In: JELIA'98. LNAI, vol. 1489, pp. 122–138. Springer (1998)
4. Bry, F., Yahya, A.: Positive unit hyperresolution tableaux and their application to minimal model generation. J. Automated Reasoning 25(1), 35–82 (2000)
5. Clarke, E.M., Schlingloff, B.: Model checking. In: Handbook of Automated Reasoning, pp. 1635–1790. Elsevier (2001)
6. Divroodi, A.R., Nguyen, L.A.: On bisimulations for description logics. CoRR abs/1104.1964 (2011)
7. Gentilini, R., Piazza, C., Policriti, A.: From bisimulation to simulation - coarsest partition problems. J. Automated Reasoning 31, 73–103 (2002)
8. Henzinger, M.R., Henzinger, T.A., Kopke, P.W.: Computing simulations on finite and infinite graphs. In: Proc. FCS-36. pp. 453–462. IEEE Computer Society (1995)
9. Hintikka, J.: Model minimization - an alternative to circumscription. J. Automated Reasoning 4(1), 1–13 (1988)
10. Lorenz, S.: A tableaux prover for domain minimization. J. Automated Reasoning 13(3), 375–390 (1994)
11. Lutz, C., Wolter, F.: Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. J. Symbolic Computation 45(2), 194–228 (2010)
12. Niemelä, I.: Implementing circumscription using a tableau method. In: Proc. ECAI '96. pp. 80–84. Wiley (1996)
13. Papacchini, F., Schmidt, R.A.: A tableau calculus for minimal modal model generation. ENTCS 278(3), 159–172 (2011)
14. Reiter, R.: A theory of diagnosis from first principles. Artificial Intelligence 32(1), 57–95 (1987)