Terminating Minimal Model Generation Procedures for Propositional Modal Logics*

Fabio Papacchini and Renate A. Schmidt

The University of Manchester, UK

Abstract. Model generation and minimal model generation are useful for tasks such as model checking and for debugging of logical specifications. This paper presents terminating procedures for the generation of models minimal modulo subset-simulation for the modal logic **K** and all combinations of extensions with the axioms **T**, **B**, **D**, **4** and **5**. Our procedures are minimal model sound and complete. Compared with other minimal model generation procedures, they are designed to have smaller search space and return fewer models. In order to make the models more effective for users, our minimal model criterion is aimed to be semantically meaningful, intuitive and contain a minimal amount of information. Depending on the logic, termination is ensured by a variation of equality blocking.

1 Introduction

Automated reasoning methods are often designed to check satisfiability and validity of formulae. In many applications the "yes or no" answer returned by these methods is all the information that is needed, but there are tasks where additional information is required. Model generation methods complement such automated reasoning methods by returning models that explain why a certain answer holds. Examples of tasks where model generation methods are useful are fault analysis, model checking and debugging of logical specifications [18, 13]. 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, which makes effective model generation a challenging problem. For these reasons, there have been several studies about the generation of different kinds of *minimal* models for classical and non-classical logics [2, 15, 14, 16, 17, 13, 7].

In this paper we introduce a semantic notion of minimality, similar to the notions used in [17, 13]. Our minimality criterion is based on a preorder on models called subset-simulation (i.e., it is a variation of the more common notion of simulation [4]). The criterion is designed so that minimal models are semantically meaningful, more natural than models minimal with respect to other minimality criteria, and contain a minimal amount of information. In this paper we propose

* The first author is supported by an EPSRC EU Doctoral Training Award. The research was partially supported by EPSRC research grant EP/H043748/1.

the first terminating, minimal model sound and complete procedures for the generation of models minimal modulo subset-simulation for all normal modal logics in between **K** and **S5**. If a model generator is minimally sound and complete, not only will it generate one minimal model, but it will generate the complete set of all minimal models. In comparison with other approaches, our procedures benefit from smaller search spaces, and fewer models are returned. In particular, we aim to return the smallest set of all relevant minimal models, so that the user is not swamped with too many similar models.

As modal logics are closely related to description logics, our procedures can be used as methods complementary with respect to ontology debugging methods such as the ones proposed in [19, 8]. The usual definition of ontology debugging assumes that an ontology is incoherent (inconsistent). In this case, debugging is the ability to identify the cause of the incoherence and fix it. An ontology can however be considered faulty even when it is coherent, but it does not properly model the domain of interest. This can be because aspects and properties of the domain of interest, that are expected to hold, do not follow from the ontology. In this context, similarly to test-driven software development paradigms, our procedures complement the notion of ontology debugging and provide assistance to model correctly the domain of interest. Minimal model generation procedures can be used to check whether these properties hold at any stage of the life cycle of the ontology, and then corrected based on the computed models.

Another possible use for the generation of models minimal modulo subset-simulation is positive query answering for Horn fragments of modal logics similar to [13]. In [13] the query answering problem is reduced to a model checking problem. Our procedures can be used in the same way, but it is not restricted to Horn fragments of modal logics.

The logics and the main properties of models minimal modulo subset-simulation are presented in Section 2. Section 3 defines minimal model sound and complete procedures for the logics under consideration. As minimal model soundness can be easily shown if the procedures are minimal model complete, we focus on formally proving minimal model completeness (Section 4). Termination results for all the logics under consideration are presented in Section 5, as one of the main contributions of the paper. How our procedures relate to other minimal model generation procedures, what are possible extensions of the procedures, and how it is possible to further improve them is discussed in Section 6. Section 7 summarises the contributions of the paper and mentions directions of future work.

2 Modal Logics and the Minimality Criterion

We work with modal formulae of the propositional modal logic **K** possibly extended with a subset of the well-known axioms **T**, **B**, **D**, **4**, and **5**. Specifically, the logics covered in this paper are all propositional normal modal logics below **S5**, namely, **K**, **KD**, **KDB**, **K4**, **K5**, **KD4**, **KD5**, **K45**, **KD45**, **KB4**,

Table 1. Modalities and their corresponding frame conditions

$\Box$	Axiom	Frame condition	First-order representation
K			
T	$\Box p \to p$	reflexivity	$\forall x R(x,x)$
B	$p \to \Box\Diamond p$	symmetry	$\forall x \forall y (R(x,y) \to R(y,x))$
D	$\Box p \to \Diamond p$	seriality	$\forall x \exists y R(x,y)$
4	$\Box p \to \Box\Box p$	transitivity	$\forall x \forall y \forall z (R(x,y) \wedge R(y,z) \to R(x,z))$
5	$\Diamond p \to \Box\Diamond p$	Euclideanness	$\forall x \forall y \forall z (R(x,y) \wedge R(x,z) \to R(y,z))$

KT4, and **KT5**(= **S5**). All these logics are decidable. Table 1 lists the axioms and their semantic characterisations as frame properties.

A *modal formula* is a formula of the form $\top$, $\bot$, p_i, $\neg\phi$, $\phi_1 \wedge \phi_2$, $\phi_1 \vee \phi_2$, $\Diamond\phi$, $\Box\phi$, where $\top$ and $\bot$ are two nullary logical operators for, respectively, true and false; $p_i \in \Sigma$ is a propositional symbol belonging to the set Σ of propositional symbols; $\neg$, $\wedge$, $\vee$, $\Diamond$, $\Box$ are, respectively, the logical operators of negation, conjunction, disjunction, diamond and box; and ϕ_1, ϕ_2, ϕ are modal formulae.

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

$$
\begin{array}{ll}
\mathcal{I}, u \not\models \bot & \mathcal{I}, u \models \top \\
\mathcal{I}, u \models p_i & \text{iff } p_i \in V(u) \\
\mathcal{I}, u \models \neg\phi & \text{iff } \mathcal{I}, u \not\models \phi \\
\mathcal{I}, u \models \phi_1 \vee [\wedge]\ \phi_2 & \text{iff } \mathcal{I}, u \models \phi_1 \text{ or[and] } \mathcal{I}, u \models \phi_2 \\
\mathcal{I}, u \models \Box\phi & \text{iff for every } v \in W \text{ if } (u,v) \in R \text{ then } \mathcal{I}, v \models \phi \\
\mathcal{I}, u \models \Diamond\phi & \text{iff there is a } v \in W \text{ such that } (u,v) \in R \text{ and } \mathcal{I}, v \models \phi
\end{array}
$$

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.

A *model graph* $M = (W, R, \mathcal{V})$ is an interpretation except that $\mathcal{V}(u)$ returns the set of formulae true in u. Given a model graph $M = (W, R, \mathcal{V})$ it is possible to obtain the corresponding interpretation $\mathcal{I} = (W, R, V)$, where $V(u) = \mathcal{V}(u) \cap \Sigma$ for all $u \in W$.

Let u and v be two elements of the domain of a model $\mathcal{I}$. If there is a path in the model from u to v, then u is an *ancestor* of v and v is a *descendant* of u.

The *frame closure* of a model $\mathcal{I}$ is the model obtained by computing the closures of the relevant frame properties (e.g., transitive closure).

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

- $V(u) = V'(u')$,
- if uRv, then there exists a $v' \in W'$ such that $u'Rv'$ and vBv', and
- if $u'R'v'$, then there exists a $v \in W$ such that uRv and vBv'.

An *auto-bisimulation* is a bisimulation between a model and itself.

Let $\mathcal{I} = (W, R, V)$ and $\mathcal{I}' = (W', R', V')$ be two models of a modal formula ϕ. A *subset-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.

- $V(u) \subseteq V'(u')$, and
- if uRv, then there exists a $v' \in W'$ such that $u'Rv'$ and vSv'.

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 $\mathcal{I}$ to $\mathcal{I}'$. 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 $\mathcal{I}$ and $\mathcal{I}'$, if there is a full subset-simulation S from $\mathcal{I}$ to $\mathcal{I}'$, we say that $\mathcal{I}'$ *subset-simulates* $\mathcal{I}$, or $\mathcal{I}$ *is subset-simulated* by $\mathcal{I}'$. We write $\mathcal{I} \leq_{\subseteq} \mathcal{I}'$ if $\mathcal{I}$ is subset-simulated by $\mathcal{I}'$.

Subset-simulation is a preorder on models. That is, subset-simulation is a reflexive and transitive relation on models. For this reason it can be used to define the following minimality criterion. A model $\mathcal{I}$ of a modal formula φ is *minimal modulo subset-simulation* iff for any model $\mathcal{I}'$ of φ, if $\mathcal{I}' \leq_{\subseteq} \mathcal{I}$, then $\mathcal{I} \leq_{\subseteq} \mathcal{I}'$.

As bisimulation is more restrictive than subset-simulation, any model $\mathcal{I}_B$ bisimilar to a model $\mathcal{I}$ preserves the original subset-simulation relationship of $\mathcal{I}$. This result is formally expressed in the following lemma, and is used for proving termination of our procedures.

Lemma 1. *Bisimulation preserves subset-simulation. That is, given two models $\mathcal{I}$ and $\mathcal{I}'$, any bisimilar model $\mathcal{I}_B$ of $\mathcal{I}$ is such that if $\mathcal{I} \leq_{\subseteq} \mathcal{I}'$ then $\mathcal{I}_B \leq_{\subseteq} \mathcal{I}'$, and if $\mathcal{I}' \leq_{\subseteq} \mathcal{I}$ then $\mathcal{I}' \leq_{\subseteq} \mathcal{I}_B$.*

3 Procedures for the Generation of Minimal Models

Our procedures for the generation of models minimal modulo subset-simulation are composed of a tableau calculus and a minimality test. Depending on which logic below **S5** is considered, different rules for handling frame properties and different termination techniques are used. The tableau calculus, without the minimality test, is devised to generate minimal models, but it can also generate non-minimal models. We prove minimal model completeness of the calculus in Section 4, and that the use of the minimality test results in minimal model sound and complete procedures.

As the minimality criterion is based on a preorder, it is possible to have symmetry classes of models and minimal models belong to the same symmetry class. Models that belong to the same symmetry class share the same positive information, meaning that all the models entail the same positive formulae. For this reason, we define minimal model completeness as follows. A procedure is

Table 2. Tableau calculus for the generation of minimal models

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

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

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

$$(\Diamond)\ \frac{u:\Diamond\phi}{\begin{array}{c} (u,v):R \\ v:\phi \end{array}}$$

where v is fresh.

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

minimal model complete if it generates at least one witness for each symmetry class of minimal models.

The input to the calculus is a modal formula in negation normal form labelled by an initial world u. Transformation to negation normal form is not essential, but it simplifies the presentation. It also means that there is no need for preprocessing before applying the calculus, and reduces the number of rules in the calculus. Disjunctions and conjunctions are assumed to be flattened (e.g., 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, $\Diamond\phi$ or $\Box\phi$. We use Φ^+ to denote a non-empty disjunction, where all disjuncts are of the form $\mathcal{A}$, and use 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 $neg(\Phi^+)=\top$. The exclusive selection of positive propositional variables is crucial for the minimal model completeness of the calculus. An example of this is given in the explanation of the (β) rule.

Table 2 presents the rules of the calculus for the modal logic **K**. Given an input formula $u:\phi$, the rules are exhaustively applied. At most one rule is applied to any formula appearing as the main premise, where the *main premise* of a multi-premise rule is the premise on the right. For fairness, each instance of a rule application is performed exactly once. Given an open branch $\mathcal{B}$ in a tableau derivation, a model $\mathcal{I}=(W,R,V)$ can be extracted from $\mathcal{B}$ as follows. The domain W is the set of all the labels occurring in $\mathcal{B}$, the accessibility relation is composed of all the instances $(u,v):R$ in $\mathcal{B}$, the interpretation function V is such that $V(u)=\{p_i \mid u:p_i\in\mathcal{B}\}$. A partial model graph M is extracted from a branch $\mathcal{B}$ in a similar way, except that $\mathcal{V}(u)=\{\phi\mid u:\phi\in\mathcal{B}\}$.

The (α) rule is a variation of standard rule for conjunctions. If $\Phi^+_\alpha = \top$ then it just expands the conjunction, otherwise the application of the (α) rule performs lazy clausification. If such 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.

The $(\Box)$ rule and the $(\Diamond)$ rule are the common rules for box and diamond formulae. They simply expand formulae in the scope of a modality as required by their semantics.

The (β) rule is the only branching rule 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 (more common uses of complement splitting can be found in the literature, e.g. [2]). 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 $u : \phi_1 \vee \Box\phi_2$. If the complement $\Diamond\neg\phi_2$ of $\Box\phi_2$ 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 (SBR) rule is a selection-based resolution rule. It can be seen as a weaker version of the (SBR) rule in [16], the $PUHR$ rule in [2], or the hyper-tableau rule in [1]. 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 (i.e., all negative propositional variables) from a disjunction. The rationale for the (SBR) rule is that if a disjunction contains negative information (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 a 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.

Theorem 1. *The tableau calculus in Table 2 is sound and refutationally complete for* **K**.

For reasons of space we omit a formal proof, but the calculus does not differ much from known calculi. All the rules are sound variations of common rules. The rule modifications help in directing the calculus toward the generation of minimal models, for example, the restrictions in Φ^+ or Φ^+_α.

In the next section we show that the calculus is minimal model complete. *Minimal model completeness* means that the calculus generates at least one

Table 3. Rules for extending the calculus

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

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

$$(\mathbf{D})\ \frac{}{u:\Diamond\top}$$

witness per symmetry class of minimal models. We also want the calculus to be *minimal model sound*, that is, only minimal models are generated.

To achieve minimal model soundness we define a minimality test to close branches from which non-minimal models can be extracted. The minimality test is called *subset-simulation test.* It consists of two operations. First, let $\mathcal{I}$ be a partial model extracted from an open branch $\mathcal{B}$. If a model $\mathcal{I}'$ such that $\mathcal{I}' \leq_{\subseteq} \mathcal{I}$ has already been found, then close $\mathcal{B}$. Second, let $\mathcal{I}$ be a model newly extracted from an open and fully-expanded branch $\mathcal{B}$. If a model $\mathcal{I}'$ such that $\mathcal{I}' \leq_{\subseteq} \mathcal{I}$ has already been found, then close $\mathcal{B}$, and for any already extracted model $\mathcal{I}'$, if $\mathcal{I} \leq_{\subseteq} \mathcal{I}'$, then close the branch from which $\mathcal{I}'$ was extracted.

Computing subset-simulation relations between finite models is a decidable problem. Our procedure uses the algorithm for computing subset-simulations presented in [17], which is a variation of the algorithm for computing autosimulation in [6]. The complexity of the algorithm depends directly on the size of the domains and on the number of relations in the involved models.

Our minimal model generation procedure extends to all sublogics of **S5**. Table 3 contains the structural rules enabling the handling of all these logics. Augmenting the extensions with the subset simulation test results in minimal model sound and complete procedures. This is because the subset-simulation test is independent of the logic. What matters is minimal model completeness, and [17] proves that such structural rules preserve minimal model completeness. It is worth noting that some of the extensions, e.g., **K4**, might have minimal models with an infinite domain, and this affects termination. However, minimal model soundness and completeness can be ensured by choosing good branch selection strategies. A suitable branch selection strategy is to always select the branch with the smallest number of labels (i.e., the branch where the extracted partial model has the smallest domain). In Section 5 we show that a better branch selection strategy can be adopted as soon as termination of the procedure is ensured.

4 Minimal Model Completeness

Because our minimal model completeness proof relies on results in [17] for the multi-modal logic $\mathbf{K}_{(m)}$ and its extensions, we recall here some definitions from [17]. A *simulation* relation between models is as a subset-simulation relation, except that the first property is $V(u) = V'(u')$. $\mathcal{I} \leq_{=} \mathcal{I}'$ denotes that $\mathcal{I}$ is simulated by $\mathcal{I}'$. The minimality criterion in [17] is as follows. A model $\mathcal{I}$ of a modal formula φ is *minimal modulo subset-simulation* iff for any model $\mathcal{I}'$ of φ, if $\mathcal{I}' \leq_{\subseteq} \mathcal{I}$, then $\mathcal{I} \leq_{\subseteq} \mathcal{I}'$ and for any model $\mathcal{I}''$ of φ belonging to the same symmetry class of $\mathcal{I}$, if $\mathcal{I}'' \leq_{=} \mathcal{I}$ then $\mathcal{I} \leq_{=} \mathcal{I}''$. The notion of minimal model completeness used in [17] requires the generation of all minimal models, and not just a witness per symmetry class.

As the logics considered in this paper are a subset of the logics considered in [17], the following lemma is restricted to the logics covered in this paper.

Lemma 2. *Let $\mathcal{I}$ and $\mathcal{I}'$ be models of a modal formula φ such that $\mathcal{I}$ is minimal with respect to the minimality criterion in [17], and $\mathcal{I}'$ is minimal with respect to the minimality criterion used in this paper. Then the following hold.*

- *$\mathcal{I}$ is minimal with respect to the minimality criterion used in this paper, and*
- *there exists a model $\mathcal{I}''$ minimal with respect to the minimality criterion used in [17] such that $\mathcal{I}'' \leq_{\subseteq} \mathcal{I}'$.*

The lemma explains the relation between the minimality criterion used in [17] and the minimality criterion used in this paper. The first point of Lemma 2 tells us that the minimality criterion used in this paper considers as minimal all the models considered minimal by the minimality criterion in [17], and potentially more than these. The second point tells us, indirectly, that the symmetry classes for the two notions are the same.

As the notion of minimal model completeness in [17] is wider than our notion, and given the relation between the two minimality criteria, the following holds.

Lemma 3. *The procedure in [17] is minimal model complete with respect to our notions of minimal model and minimal model completeness.*

From a procedural perspective, the minimal model generation procedure we propose and the procedure proposed in [17] mainly differ in how diamond formulae are expanded. This is due to the use of different minimality criteria and different notions of minimal model completeness, which force the calculus in [17] to explore all possible expansions of diamond formulae. The $(\Diamond)$ rule, simplified to the uni-modal case, used in [17] is the following.

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

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

Theorem 2. *For any model $\mathcal{I}'$ extracted from an open and fully expanded branch of the procedure in [17], there is a model $\mathcal{I}$ extracted from an open and fully expanded branch $\mathcal{B}$ of our procedure such that $\mathcal{I} \leq_{\subseteq} \mathcal{I}'$.*

Proof. Let $\mathcal{I} = (W, R, \mathcal{V})$ and $\mathcal{I}' = (W', R', \mathcal{V}')$. We prove the theorem inductively by creating a relation $S \subseteq W \times W'$ during the construction of the branch $\mathcal{B}$, and show that S is a full subset-simulation.

Base case: Let us assume that the input of the procedure in [17] is $u' : \varphi$, and the input of our procedure is $u : \varphi$. This means that $\mathcal{I}', u' \models \varphi$, and the initial partial model graph $\mathcal{I}$ is $(\{u\}, \emptyset, \mathcal{V}(u) = \{\varphi\})$. As $\mathcal{I}'$ is a complete model graph and $\mathcal{I}', u' \models \varphi$, then $\varphi \in \mathcal{V}'(u')$. This implies that $\mathcal{V}(u) \subseteq \mathcal{V}'(u')$. Let $(u, u') \in S$. Then it is immediate that S is a full subset-simulation from $\mathcal{I}$ to $\mathcal{I}'$.

Induction step: Let us assume that after n rule applications, for the extracted model $\mathcal{I}$ there is a subset-simulation S from $\mathcal{I}$ to $\mathcal{I}'$. We prove that S, or a variation of it, is still a subset-simulation relation after the application of any rule ρ of our procedure. For reasons of space we give proofs only for three of the rules, but all the other cases are similar. In all the following cases, we assume $\mathcal{I} = (W, R, \mathcal{V})$ is the model extracted before the application of ρ.

ρ is the (α) rule. This means that the expanded formula is a labelled disjunction $u : \varphi$, where at least one disjunct φ_α is a conjunction. Let Φ be the set of labelled formulae representing the conclusion of the (α) rule and $\Psi = \{\psi \mid u : \psi \in \Phi\}$. This means that Φ is on the branch and the new extracted model graph $\mathcal{I}'' = \mathcal{I}$, where $\mathcal{V}'' = \mathcal{V}$ except for $\mathcal{V}''(u) = \mathcal{V}(u) \cup \Psi$. By the inductive hypothesis, there is a $u' \in W'$ such that $(u, u') \in S$, $\mathcal{V}(u) \subseteq \mathcal{V}'(u')$ and $\mathcal{I} \leq_\subseteq \mathcal{I}'$. As $\varphi \in \mathcal{V}'(u')$ and $\mathcal{I}'$ is a complete model, then $\Psi \subseteq \mathcal{V}'(u')$. This implies that $\mathcal{V}''(u) \subseteq \mathcal{V}'(u')$. That is, the current S is a full subset-simulation such that $\mathcal{I}'' \leq_\subseteq \mathcal{I}'$. In principle there may be more than one conjunction to be selected for the application of the (α) rule. This implies that Φ may be different from the application of the (α) rule applied to generate $\mathcal{I}'$. Even though the two sets of conclusions are syntactically different, they are semantically equivalent. Hence, w.l.o.g. we can assume that the same conjunction is used.

ρ is the $(\Diamond)$ rule. This means that the expanded formula is a labelled diamond formula, let us say $u : \Diamond\varphi$. As a result of the application of the $(\Diamond)$ rule, $\{v : \varphi, (u, v) : R\}$ are on the branch and v is fresh on the branch. The new extracted model $\mathcal{I}''$ is as follows. $W'' = W \cup \{v\}$, $R'' = R \cup \{(u, v)\}$, and $\mathcal{V}'' = \mathcal{V}$ except for $\mathcal{V}''(v) = \{\varphi\}$. By the inductive hypothesis, there is a $u' \in W'$ such that $(u, u') \in S$, $\mathcal{V}(u) \subseteq \mathcal{V}'(u')$ and $\mathcal{I} \leq_\subseteq \mathcal{I}'$. As $\Diamond\varphi \in \mathcal{V}'(u')$ and $\mathcal{I}'$ is a complete model, then there is an R-successor $v' \in W'$ of u' such that $\varphi \in \mathcal{V}'(v')$. Let $S' = S \cup \{(v, v')\}$. S' is a full subset-simulation such that $\mathcal{I}'' \leq_\subseteq \mathcal{I}'$.

ρ is the $(\mathbf{4})$ rule. This means that there are two labelled relations $(u, v) : R$ and $(v, w) : R$ for which transitivity has not been applied yet. As a result of the application of the $(\mathbf{4})$ rule, $(u, w) : R$ is on the branch and the new extracted model $\mathcal{I}''$ is such that $\mathcal{I}'' = \mathcal{I}$, except for $R'' = R \cup \{(u, w)\}$. By the inductive hypothesis, there are $u', v', w' \in W'$ such that $(u, u'), (v, v'), (w, w') \in S$, $(u', v'), (v', w') \in R$ and $\mathcal{I} \leq_\subseteq \mathcal{I}'$. As $\mathcal{I}'$ is a complete model and R is transitive, then $(u', w') \in R$. That is, the current S is a full subset-simulation such that $\mathcal{I}'' \leq_\subseteq \mathcal{I}'$. □

Corollary 1. *For any model $\mathcal{I}'$ minimal with respect to [17], there is a model $\mathcal{I}$ generated by our procedure such that $\mathcal{I} \leq_\subseteq \mathcal{I}'$.*

Minimal model completeness of our tableau calculus follows from Lemma 3 and Corollary 1. Minimal model soundness is the result of applying the subset-simulation test to minimal model complete tableaux calculi.

Theorem 3. *The tableau calculus in Table 2 and the extensions with the rules in Table 3 are minimal model complete. That is, the calculi generate at least one witness for each symmetry class of minimal models.*

Theorem 4. *Augmenting the tableau calculus in Table 2 and its extensions with the rules in Table 3 and the subset-simulation test gives us minimal model sound and complete procedures. That is, only minimal models and at least a witness for each symmetry class of minimal models are generated.*

5 Ensuring Termination

The presented calculus is (strongly) terminating for the modal logic **K** and its reflexive and symmetric extensions (i.e., it terminates for **KT**, **KB** and **KTB**). It is known that it is always possible to generate finite models for these logics without using any termination technique. As a reference for this, [16] proves that these logics have finite minimal Herbrand models and presents a tableau calculus that does not require any termination technique.

The same reasoning cannot be used for the other normal modal logics, namely, **KD**, **KDB**, **K4**, **K5**, **KD4**, **KD5**, **K45**, **KD45**, **KB4**, **KT4**, and **KT5**. The main challenge to obtain terminating procedures for these logics is to find blocking techniques preserving minimal model completeness.

For **KD** and **KDB** it is not difficult to achieve termination while preserving minimal model completeness. This is because the seriality condition is what affects termination, forcing all models to have paths where a world in which the only true formula is $\top$ is repeated infinitely many times. It is, therefore, enough to add a reflexive edge as soon as the first such world appears, and the resulting finite model is bisimilar to the original model. Therefore, the following holds.

Theorem 5. *Our procedures to handle the modal logics* **KD** *and* **KDB** *are minimal model sound and complete, and terminate if a reflexive loop is added to each occurrence of a fully-expanded world u where $\mathcal{V}(u) = \emptyset$.*

The rule application order is important in a practical implementation. Specifically, the $(\Diamond)$ rule needs to be the last rule to be applied.

More interesting are the logics **K4**, **KT4**, and **KD4**. For these logics, the models can be infinite due to the seriality axiom or because of transitivity. The previous termination strategy is not sufficient for these logics (i.e., it is possible to have infinite chains where no fully-expanded world has an empty interpretation). The usual method to obtain a terminating tableau calculus, see, e.g. [9], is to use static subset blocking. Formally, a world u is *subset blocked* if there is a parent v of u such that $\mathcal{V}(u) \subseteq \mathcal{V}(v)$. This kind of blocking, however, is not compatible with our minimality criterion because it might potentially merge a world with less positive information and a world with more positive information. This may lead to non-minimal models being considered minimal, affecting the minimal model soundness of the procedure, or to the non-generation of some minimal model, affecting the minimal model completeness of the procedure. It turns out that

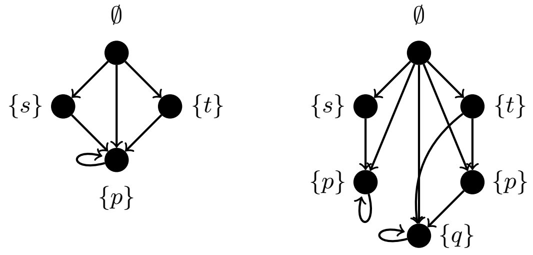

Fig. 1. Example of incompleteness when anywhere equality blocking is used

equality blocking is more suitable. However, not all forms of equality blocking can be used, but *ancestor equality blocking* can. A world u is considered *ancestor equality blocked* if there is an ancestor v of u such that $\mathcal{V}(u) = \mathcal{V}(v)$. We show that anywhere equality blocking (i.e., v can block u even if it is not an ancestor of u) violates minimal model completeness by means of an example. Suppose that $\phi = \Diamond s \wedge \Diamond t \wedge \Box\Diamond(p \vee q)$ is the input formula. While the model on the left in Figure 1 is generated when anywhere equality blocking is used, the model on the right is not and it is a model minimal modulo subset-simulation. However:

Theorem 6. *Our procedures to handle the modal logics* **K4**, **KT4** *and* **KD4** *are minimal model sound and complete, and terminate when ancestor equality blocking is used.*

Proof. Termination follows from known results that ancestor equality blocking is enough to ensure termination for these logics (e.g., [9]). We prove that minimal model completeness is preserved only for the case of **K4**, but the same reasoning can be used to prove it for the other logics. Suppose ϕ is the input modal formula. Let us assume that M is an infinite model graph of ϕ such that $M, u \models \phi$. Hence, there is at least one connected component in the graph with a path composed of an infinite sequence of worlds. Since the tableau calculus only propagates subformulae of ϕ, there are only finitely many formulae per world. This implies that the infinite path must contain a finite number of distinguishable worlds, that can be identified with each other. Due to transitivity each world in the path is connected with all its descendants, meaning we can focus our attention on the infinite repetition of a finite path of distinct worlds. This implies that M subset-simulates another model where a loop over the finite path is created. This reasoning can be iterated for all the infinite components of the graph. Hence, M subset-simulates a finite model. Therefore, as ancestor equality blocking blocks the infinite path at the first appearance of the first repeated world, minimal model completeness is preserved. □

As in the previous case, the rule application order is important. Specifically, the ($\Diamond$) rule needs to be applied last. In this way it is possible to build the model by exhaustively expanding all formulae world by world. If ancestor equality blocking is checked only before applying the ($\Diamond$) rule, then when equality is detected the set of formulae true in a world cannot change anymore.

For all the remaining cases *anywhere dynamic equality blocking* can be used. This blocking technique blocks a world if there is another world for which the interpretation is the same. The dynamic part is due to the possibility of having false guesses. Specifically, it is possible that two worlds are considered to be the same at some point in the derivation, but they do not have the same interpretation in a subsequent point in the derivation. Making the blocking technique dynamic allows for the possibility of blocking and unblocking pairs of worlds. This technique, or a variation of it, is already used in the literature (e.g., [3, 10]).

Theorem 7. *Our procedures to handle the modal logics* **K5***,* **KD5***,* **K45***,* **KD45***,* **KB4** *and* **KT5** *are minimal model sound and complete, and terminate when anywhere dynamic equality blocking is used.*

Proof. As for Theorem 6, we only need to prove that minimal model completeness is guaranteed. We prove it only for the case of **K5**, but the same reasoning can be used to prove it for the other logics. Due to frames being Euclidean, the resulting model is a strongly connected graph where the only exception is the root world. Hence, any two non-root worlds u and v are related by a reflexive, symmetric and transitive relation. If $\mathcal{V}(u) = \mathcal{V}(v)$, then an auto-bisimulation of the model would merge them into a single world. This implies that the application of anywhere dynamic equality blocking is equivalent to a bisimulation step. For Lemma 1, if the original model was minimal modulo subset-simulation, then the resulting model is still minimal modulo subset-simulation. □

From minimal model completeness and the termination results of this section, this theorem follows.

Theorem 8. *All the normal modal logics between* **K** *and* **KT5** *have finitely many symmetry classes of models minimal modulo subset-simulation.*

Having strong termination techniques for all the normal modal logics allows us to vary the branch selection strategy. Specifically, a depth-first left-to-right branch selection strategy can be used. From a practical perspective this is important because it allows for memory efficient implementations.

6 Related Work and Discussion

The most similar approach for the generation of minimal models, for both the methodology used and the minimality criterion involved, is [17]. [17] is the first paper to introduce the notion of models minimal modulo subset-simulation and, hence, first to propose a technique for the generation of such minimal models for the multi-modal logics $\mathbf{K}_{(m)}$ and some of its extensions. Our procedures have however much smaller search spaces than those in [17]. This is because our notion of minimal model completeness requires the generation of one witness per symmetry class, while the notion used in [17] requires the whole symmetry classes to be generated. This is reflected in the rules of the two tableaux calculi. The only branching rule of our tableau calculus is the (β) rule, while in [17] also the ($\Diamond$) rule is a branching rule. The ($\Diamond$) rule in [17] has a high branching factor (i.e., the number of branches is equal to current number of labels in the branch plus one), and it leads to the generation of many similar and, therefore, unnecessary models. In the literature, other notions of minimal model completeness similar to the one we adopt exist. For example, in [5, 13] not all minimal models are generated, but only witnesses of a specific kind of equivalences classes are generated.

The used notions of minimality and minimal model completeness allowed us to simplify the subset-simulation test in [17]. The subset-simulation test can be improved even more if for any extracted minimal model the auto-bisimulation is computed. This is because the complexity of checking subset-simulation relations depends on the number of worlds and the number of edges. Using auto-bisimulation can potentially result in minimal models having a smaller domain, making the comparison with other models easier. It is important to note though that the procedure proposed in [17] is designed to cover more expressive modal logics than what we cover in this paper, but, as long as termination is not taken into consideration, our approach can easily be extended to cover exactly the same expressive multi-modal logics while maintaining the results of this paper.

The minimality criterion in [13] has similarities to our minimality criterion and, using our terminology, it can be defined as a minimality criterion based on subset-bisimulation. [13] proposes a method to reduce the problem of answering positive queries for Horn modal formulae to the task of model checking. The creation of a minimal model that preserves all positive entailments simplifies the model checking task. It is interesting to note that any model minimal modulo subset-simulation is also minimal with respect to the minimality criterion proposed in [13]. This means that our approach can be used to address exactly the same problem, even for formulae that are outside of the modal Horn fragment.

Apart from our notion of minimal models, other minimality criteria exist. These can be classified into: syntactic notions of minimality, minimal Herbrand models [2, 14, 16], and domain minimality [7, 11]. The class of minimal 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 [2, 14] and for modal logics in [16]. Despite the use of a different minimality criterion, there are similarities between the models considered minimal by our approach and those considered minimal in [16]. As long as termination is not taken into consideration, models minimal modulo subset-simulation are a subset of minimal Herbrand models. As our minimality criterion takes into consideration the semantics of models, some minimal Herbrand model can be considered redundant or not minimal, resulting in a smaller set of minimal models. As soon as termination techniques are necessary, comparing the two notions of minimality becomes more difficult. This is also due to the fact that the approach proposed in [16] cannot be extended easily to cover logics with potentially infinite models, and is restricted to the multi-modal logics $\mathbf{K}_{(m)}$, $\mathbf{KT}_{(m)}$, $\mathbf{KB}_{(m)}$ and $\mathbf{KTB}_{(m)}$.

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 [7, 11]. Domain minimal models, however, tend to be counter-intuitive because too many worlds are collapsed into a single world. As a result, all the information needed to satisfy the input formula is pushed to the least number of domain elements, making tasks such as verification and debugging harder. Our approach is designed to avoid the creation of domain minimal models while spreading the positive information as much as possible. This results in more meaningful and intuitive models, as is shown in [17].

Description logics are closely related to the modal logics considered in this paper, and all results can be transferred to the corresponding description logics. An important difference is the presence of TBoxes in description logics. This difference can be accommodated by using a calculus for modal logics extended with rules for handling universal modalities. As TBoxes do not need the complete expressiveness of the universal modalities, we can extend our procedures in such a way that only specific patterns of universal modalities are allowed. In this way the procedures can handle description logics such as $\mathcal{ALC}$ with non-empty TBoxes. ABoxes pose no technical challenges. The full expressive power of universal modalities, however, increases the complexity of the procedure, and termination techniques preserving minimal model completeness are needed.

In this paper we used structural rules to accommodate frame conditions. A common alternative to the structural rules are propagation rules (e.g., [12]). The use of propagation rules is possible, but it would require expensive changes to the procedures. It can be proved that if there is a subset-simulation relation between two models obtained by using propagation rules, then the same subset-simulation relation holds also for the frame closures of the models. As the complexity of computing subset-simulation depends on the size of the domain and on the number of edges, the use of propagation rules seems promising. On the other hand, the other direction does not hold. In particular, subset-simulation relations between models where the frame closures are computed are not necessarily transferred to models generated by using a procedure based on propagation rules. As the subset-simulation test is applied many times, the use of propagation rules would require repeated computations of frame closures leading to worse performance.

7 Conclusion

We presented the first terminating, minimal model sound and complete procedures for the generation of models minimal modulo subset-simulation for all the sublogics of **S5**. Compared with other minimal model generation approaches, our procedures greatly benefit from smaller search spaces, fewer models are generated, and the semantically meaningfulness and naturalness of the models make them more effective for debugging purposes. These features of the procedures are really promising from both an implementation and a practical point of view.

We plan to extend our procedures by introducing rules handling more expressive modal logics. Logics we aim to handle are all the extensions from the uni-modal case to the multi-modal case, converse relations, universal modalities and inclusion axioms. These generalisations correspond to expressive logics widely used in real world applications. This is why we are currently working on implementing the procedures. We believe efficient implementations are achievable, and they will have important impact by complementing and improving techniques for debugging and verification.

References

1. Baumgartner, P., Fürbach, U., Niemelä, I.: Hyper tableaux. In: Proc. JELIA'96. LNCS, vol. 1126, pp. 1–17. Springer (1996)
2. Bry, F., Yahya, A.: Positive unit hyperresolution tableaux and their application to minimal model generation. J. Automat. Reason. 25 (1), 35–82 (2000)
3. Cialdea Mayer, M.: A proof procedure for hybrid logic with binders, transitivity and relation hierarchies. In: Proc. CADE'13. LNCS, vol. 7898, pp. 76–90. Springer (2013)
4. Clarke, E.M., Schlingloff, B.: Model checking. In: Robinson, A., Voronkov, A. (eds.) Handbook of Automated Reasoning, pp. 1635–1790. Elsevier (2001)
5. Denecker, M., De Schreye, D.: On the duality of abduction and model generation in a framework for model generation with equality. Theoret. Computer Sci. 122 (1&2), 225–262 (1994)
6. Henzinger, M.R., Henzinger, T.A., Kopke, P.W.: Computing simulations on finite and infinite graphs. In: Proc. FCS-36. pp. 453–462. IEEE Comput. Soc. (1995)
7. Hintikka, J.: Model minimization—An alternative to circumscription. J. Automat. Reason. 4 (1), 1–13 (1988)
8. Horridge, M., Parsia, B., Sattler, U.: Extracting justifications from bioportal ontologies. In: Proc. ISWC'12 (2). LNCS, vol. 7650, pp. 287–299. Springer (2012)
9. Horrocks, I., Hustadt, U., Sattler, U., Schmidt, R.A.: Computational modal logic. In: Blackburn, P., van Benthem, J., Wolter, F. (eds.) Handbook of Modal Logic, pp. 181–245. Elsevier (2007)
10. Horrocks, I., Sattler, U.: A description logic with transitive and inverse roles and role hierarchies. J. Logic Comput. 9 (3), 385–410 (1999)
11. Lorenz, S.: A tableaux prover for domain minimization. J. Automat. Reason. 13 (3), 375–390 (1994)
12. Massacci, F.: Single step tableaux for modal logics. J. Automat. Reason. 24 (3), 319–364 (2000)
13. Nguyen, L.A.: Constructing finite least Kripke models for positive logic programs in serial regular grammar logics. Logic J. IGPL 16 (2), 175–193 (2008)
14. Niemelä, I.: Implementing circumscription using a tableau method. In: Proc. ECAI'96. pp. 80–84. Wiley (1996)
15. Niemelä, I.: A tableau calculus for minimal model reasoning. In: Proc. TABLEAUX'96. LNCS, vol. 1071, pp. 278–294. Springer (1996)
16. Papacchini, F., Schmidt, R.A.: A tableau calculus for minimal modal model generation. Electr. Notes Theoret. Computer Sci. 278 (3), 159–172 (2011)
17. Papacchini, F., Schmidt, R.A.: Computing minimal models modulo subset-simulation for propositional modal logics. In: Proc. FroCoS'13. LNAI, vol. 8152, pp. 279–294. Springer (2013)
18. Reiter, R.: A theory of diagnosis from first principles. Artificial Intelligence 32 (1), 57–95 (1987)
19. Schlobach, S., Huang, Z., Cornet, R., van Harmelen, F.: Debugging incoherent terminologies. J. Automat. Reason. 39 (3), 317–349 (2007)