A Tableau Method for Checking Rule Admissibility in S4

Sergey Babenyshev[a], Vladimir Rybakov[a], Renate A. Schmidt[b] and Dmitry Tishkovsky[b]

[a] *Department of Computing and Mathematics, Manchester Metropolitan University, UK*

[b] *School of Computer Science, The University of Manchester, UK*

Abstract

Rules that are admissible can be used in any derivations in any axiomatic system of a logic. In this paper we introduce a method for checking the admissibility of rules in the modal logic S4. Our method is based on a standard semantic ground tableau approach. In particular, we reduce rule admissibility in S4 to satisfiability of a formula in a logic that extends S4. The extended logic is characterised by a class of models that satisfy a variant of the co-cover property. The class of models can be formalised by a well-defined first-order specification. Using a recently introduced framework for synthesising tableau decision procedures this can be turned into a sound, complete and terminating tableau calculus for the extended logic, and gives a tableau-based method for determining the admissibility of rules.

Keywords: Tableau calculus, admissible rule, modal logic, S4, tableau synthesis framework.

1 Introduction

Logical admissible rules were first considered by Lorenzen [12]. Initial investigations were limited to observations on the existence of interesting examples of admissible rules that are not derivable (see Harrop [7], Mints [13]). The area gained new momentum when Friedman [2] posed the question whether algorithms exist for recognising whether rules in intuitionistic propositional logic IPC are admissible. This problem and the corresponding problem for modal logic S4 are solved affirmatively in Rybakov [15,17]. The same approach can be used for a broad range of propositional modal logics, for example K4, S4, GL [18]. Roziere [14] presents a solution to Friedman's problem for IPC that uses methods of proof theory.

The theory of admissible rules in S4 does not have the finite approximation property [11] in the strict sense. The algorithm in [15] is based on the existence of a model (of bounded size) that witnesses the non-admissibility of a rule, but is not necessarily a model for all the other admissible rules. In [1] it is observed that a witnessing model can be obtained by filtration.

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.com/locate/entcs

Admissibility of rules has direct connections to the problem of unification. A refined technique [18] is used for admissibility of rules with meta-variables, for unification, for unification with parameters, and for the solvability of logical equations in transitive modal logics.

Algorithms deciding admissibility for some transitive modal logics and IPC, based on projective formulae and unification, are described in Ghilardi [3,4,5,6]. They combine resolution and tableau approaches for finding projective approximations of a formula and rely on the existence of an algorithm for theorem proving. A practically feasible realisation for S4 built on the algorithm for IPC in [5] is described in [24]. These algorithms were specifically designed for finding general solutions for matching and unification problems. In contrast, the original algorithm of [15] can be used to find only *some* solution of such problems in S4.

In this paper we focus on S4 and introduce a new method for recognising the admissibility of rules. Our method is based on the reduction of the problem of rule admissibility in S4 to the satisfiability of a certain formula in an extension of S4. We refer to the extended logic as $\mathsf{S4}^{\mathsf{u},+}$. $\mathsf{S4}^{\mathsf{u},+}$ can be characterised by a class of models, which satisfy a variant of the co-cover property that is definable by modal formulae. This property is expressible in first-order logic and means that the semantics of the logic can be formalised in first-order logic. We exploit this property in order to devise a tableau calculus for $\mathsf{S4}^{\mathsf{u},+}$ in a recently introduced framework for automatically synthesising tableau calculi and decision procedures [21,22].

The tableau synthesis method of [22] works as follows.

(i) The user defines the formal semantics of the given logic in a many-sorted first-order language so that certain well-definedness conditions hold.

(ii) The method automatically reduces the semantic specification of the logic to Skolemised implicational forms which are then rewritten as tableau inference rules. These are combined with some default closure and equality rules.

The set of rules obtained in this way provides a sound and constructively complete calculus. Furthermore, this set of rules automatically has a subformula property with respect to a finite subformula operator. If the logic can be shown to admit finite filtration with respect to a well-defined first-order semantics then adding a general blocking mechanism produces a terminating tableau calculus [21].

We show how, using this method, a sound, complete and terminating tableau calculus can be synthesised for the extended logic $\mathsf{S4}^{\mathsf{u},+}$. This tableau calculus is then incorporated into a method for solving the rule admissibility problem in S4.

The paper is structured as follows. In Section 2 we define the syntax and semantics of the modal logic S4, and its extension $\mathsf{S4}^{\mathsf{u}}$ with the universal modality, in such a way that they can be accommodated in the tableau synthesis framework of [22]. The section also defines standard modal logic constructions and notions required for the main results of the paper. In Section 3, we give definitions of derivable and admissible rules for S4 and state Rybakov's criterion for testing admissibility of rules in S4. The reduction of rule admissibility in S4 to satisfiability in $\mathsf{S4}^{\mathsf{u},+}$ is described in Section 4. In Section 5, we show how the formulaic variant of the co-cover property can be expressed by a set of first-order formulae that provide a suitable background theory for the specification of the semantics of $\mathsf{S4}^{\mathsf{u},+}$ in the

Definitions of connectives	Background theory
$\forall x\ \big(\nu(\neg p,x) \leftrightarrow \neg\nu(p,x)\big)$	$\forall x,y,z\ \ \big(R(x,y)\wedge R(y,z) \rightarrow R(x,z)\big)$
$\forall x\ \big(\nu(p\vee q,x) \leftrightarrow \nu(p,x)\vee\nu(q,x)\big)$	$\forall x\ \ R(x,x)$
$\forall x\ \big(\nu(\Diamond p,x) \leftrightarrow \exists y\ (R(x,y)\wedge\nu(p,y))\big)$	

Figure 1. Specification of the semantics of $\mathsf{S4}$ in S_{S4}

tableau synthesis framework. Within this framework we can then synthesise sound and complete tableau calculi for the logics $\mathsf{S4}^{\mathsf{u},+}$ and $\mathsf{S4}^{\mathsf{u}}$. Under certain conditions it is possible to refine the rules of the calculus that are generated by default [22]. These conditions are true for the logics we consider and is discussed in Section 6. Section 7 describes how terminating tableau calculi can be obtained by adding the unrestricted blocking mechanism of [20,21]. In Section 8, we finally give an algorithm for testing rule admissibility, but also rule derivability, in $\mathsf{S4}$. In Section 9 we conclude with a discussion of the applicability of the algorithm and method to other logics and various problems closely related to admissibility.

For lack of space some of the details of the tableau synthesis framework are omitted; for these the interested reader is referred to [22] and also [20,21].

2 Syntax and Semantics

We denote the language of the modal logic $\mathsf{S4}$ by $\mathcal{L}_{\mathsf{S4}}$. $\mathcal{L}_{\mathsf{S4}}$ is given by the standard modal language over a countable set of propositional variables $\{p, q, p_0, q_0, \ldots\}$, the Boolean logical connectives $\neg$ and $\vee$, and the modal connective $\Diamond$. Other Boolean connectives such as $\top$, $\bot$, $\wedge$, $\rightarrow$, and $\leftrightarrow$ and the modal connective $\Box$ are defined via $\neg$, $\vee$, and $\Diamond$ as usual.

Let $\mathcal{L}^{\mathsf{u}}_{\mathsf{S4}}$ denote the extension of the language $\mathcal{L}_{\mathsf{S4}}$ with the 'somewhere' modality $\langle\mathsf{u}\rangle$. The dual modality $\neg\langle\mathsf{u}\rangle\neg$ is the universal modality, which is denoted by $[\mathsf{u}]$.

For a formula ϕ, we denote by $\mathsf{sub}(\phi)$ the set of all subformulae of ϕ. Let $\mathsf{sub}(\Sigma) \overset{\mathsf{def}}{=} \bigcup\{\mathsf{sub}(\phi) \mid \phi \in \Sigma\}$ for any set of formulae Σ. We usually write $\mathsf{sub}(\phi_1, \ldots, \phi_n)$ rather than $\mathsf{sub}(\{\phi_1, \ldots, \phi_n\})$. A set of formulae Σ is called a *signature* iff $\mathsf{sub}(\Sigma) = \Sigma$.

Following the tableau synthesis framework in [22], we accommodate $\mathcal{L}^{\mathsf{u}}_{\mathsf{S4}}$ in a multi-sorted first-order specification language. This specification language includes a countable set $\{x, y, z, x_0, y_0, z_0, \ldots\}$ of first-order variables that represent elements in models, the binary predicate symbol R of the background theory that represents the accessibility relation, and a binary (intersort) predicate symbol ν which represents the forcing relation $\models$. ν can be thought of as a holds predicate. Figure 1 gives the first-order semantic specification of the standard semantics of $\mathsf{S4}$ in the framework. We denote the specification by S_{S4}. In addition to the formulae of Figure 1, S_{S4} includes the usual congruence axioms for the equality predicate $\approx$, see [22] for details.

Thus, an $\mathsf{S4}$-*(Kripke) model* (of a signature Σ) is a first-order model $M = \langle W^M, R^M, \nu^M\rangle$ that validates all the formulae of Figure 1 under arbitrary substitutions of $\mathcal{L}_{\mathsf{S4}}$-formulae (from the signature Σ) for variables p and q.

Let $\mathsf{S4}^{\mathsf{u}}$ be the extension of $\mathsf{S4}$ with the somewhere (or universal) modality. The semantic specification of $\mathsf{S4}^{\mathsf{u}}$, denoted by $S_{\mathsf{S4}^{\mathsf{u}}}$, is the extension of the specification S_{S4} of $\mathsf{S4}$ with

$$\forall x\,(\nu(\langle\mathsf{u}\rangle p, x) \leftrightarrow \exists y\,\nu(p, y)). \tag{1}$$

This defines the somewhere modality $\langle\mathsf{u}\rangle$. An $\mathsf{S4}^{\mathsf{u}}$-model (of a signature Σ) is by definition a first-order model $M = \langle W^M, R^M, \nu^M\rangle$ that validates this formula in addition to all the formulae of Figure 1 under arbitrary substitutions of $\mathcal{L}^{\mathsf{u}}_{\mathsf{S4}}$-formulae (from the signature Σ) for propositional variables p and q.

Note that if M is a model of the signature Σ then M is also a model of any signature $\Sigma' \subseteq \Sigma$. In the other direction, it is clear that every model M of a smaller signature Σ' can be extended to a model of a bigger signature Σ by (re)defining the interpretation of ν on formulae from $\Sigma \setminus \Sigma'$ (by induction on their length). We often use these facts when defining models.

A *(Kripke) frame* of $\mathsf{S4}$ (resp. $\mathsf{S4}^{\mathsf{u}}$) is a first-order structure $F = \langle W^F, R^F\rangle$ that validates all the formulae of the background theory of S_{S4} (resp. $S_{\mathsf{S4}^{\mathsf{u}}}$). A model M is *based on a frame* F (or the *underlying frame* of M is F) iff $W^M = W^F$ and $R^M = R^F$.

A *cluster* of a model M is a set $U \subseteq W^M$ such that for all $w \in W^M$ and $u \in U$ it holds that $(w, u), (u, w) \in R^M$ iff $w \in U$. Any set of R^M-incomparable clusters of M is called an *anti-chain* in M. A cluster $U \subseteq W^M$ is *maximal* iff $(u, w) \in R^M$ implies $w \in U$ for all $w \in W^M$ and $u \in U$.

Let Σ be a fixed signature and M a model of the signature Σ. We say that a formula $\phi \in \Sigma$ is *true* (*satisfied*) in a world $w \in W^M$ (in symbols $M, w \models \phi$) iff $(\phi, w) \in \nu^M$. A formula ϕ (from Σ) is *valid* in M (in symbols $M \models \phi$) iff $M, w \models \phi$ for every $w \in W^M$. And, as usual, ϕ is *satisfiable* in M iff there is $w \in W^M$ such that $M, w \models \phi$. A formula $\phi \in \Sigma$ is *valid* in a frame F (in symbols $F \models \phi$) iff it is true in every model M of the signature Σ based on F.

For a signature Σ and every element w of W^M we define a Σ-*type* $\tau^\Sigma(w)$ of w as the set of all formulae from Σ which are true in w, namely:

$$\tau^\Sigma(w) \overset{\mathsf{def}}{=} \{\psi \in \Sigma \mid M, w \models \psi\}.$$

We omit the superscript Σ and write $\tau(w)$ if Σ is known from the context. A model M of a signature Σ is called Σ-*differentiated* iff $\tau^\Sigma(w) = \tau^\Sigma(v)$ implies $w = v$ for all $w, v \in W^M$. A model M is *formulaic* iff for every element $w \in W^M$ there is a formula ϕ such that $M, w \models \phi$ and $M, v \not\models \phi$ for all $v \in W^M \setminus \{w\}$. It is clear that if Σ is finite then every Σ-differentiated model is also formulaic.

We call a Kripke model M of a signature Σ' a *definable variant* of a model N of a signature Σ if M and N are based on the same frame and for every propositional variable $p \in \Sigma'$ there is a formula $\phi \in \Sigma$ such that $M, w \models p \iff N, w \models \phi$ for every $w \in W^M = W^N$.

Let Σ be the set of all formulae in n variables $p_1, \ldots, p_n$. An $\mathsf{S4}$-model M of signature Σ is called an *n-characterising* model for $\mathsf{S4}$ iff $M \models \phi \iff \phi \in \mathsf{S4}$ for every $\phi \in \Sigma$.

3 Rules Admissible for S4

Since the language of S4 contains conjunction, without loss of generality we consider only one-premise rules. A *rule* is a pair $\langle\alpha,\beta\rangle$ of $\mathcal{L}_{\mathsf{S4}}$-formulae, usually written α/β. A rule $r=\alpha/\beta$ is *valid* in a model M (written $M\models r$) iff $M\models\alpha$ implies $M\models\beta$. A rule r is *valid* on a frame F (written $F\models r$) iff r is valid in any model M based on F. Two rules r_1, r_2 are *semantically equivalent*, or simply *equivalent*, if $F\models r_1 \iff F\models r_2$ for any frame F.

A rule $r=\alpha/\beta$ is *derivable* in S4 iff β is derivable from α and the theorems of S4 with the rule of modus ponens and the rule of necessitation. It is clear that r is derivable in S4 iff r is valid in every S4-model. The following variant of the deduction theorem in S4 can be proved by standard methods.

Theorem 3.1 *A rule α/β is derivable in* S4 *iff* $[\mathsf{u}]\alpha\wedge\neg\beta$ *is unsatisfiable in* $\mathsf{S4}^u$.

A rule $r=\alpha/\beta$ is *admissible* for the modal logic S4, written $r\in\mathsf{Adm}(\mathsf{S4})$, if for every substitution σ from $\sigma(\alpha)\in\mathsf{S4}$ it follows that $\sigma(\beta)\in\mathsf{S4}$.

A series $\mathsf{Ch}_n(\mathsf{S4})$, $n>0$, of formulaic and n-characterising S4 models is described in [18]. These models are used in the description of the following admissibility criterion.

Theorem 3.2 (Corollary of Theorem 3.3.3 [18]) *A rule is admissible in* S4 *iff it is valid in a definable variant of* $\mathsf{Ch}_n(\mathsf{S4})$ *for each* $n>0$.

The most important property for the result of this paper is that the models $\mathsf{Ch}_n(\mathsf{S4})$, $n>0$, and their definable variants possess the co-cover property. That the admissible rules for S4 can be characterized by the class of models with the co-cover property is shown in [16]. A more general result appears in [18, Theorem 3.5.1]. Similar characterisation results for IPC appear in [9, Theorem 4.1(iv)] and [10, Corollary 3.14].

By definition, a model M has the *co-cover property* (CCP) if for every finite anti-chain $D\subseteq W^M$ (D may be empty), there exists a one-element cluster $\{w\}\subseteq W^M$ such that

$$\{u\in W^M \mid (w,u)\in R^M\}=\{w\}\cup\bigcup_{v\in D}\{u\in W^M\mid (v,u)\in R^M\}.$$

The one-element cluster is called a *co-cover for D*. Note that a co-cover for the empty anti-chain is a maximal one-element cluster of M.

A rule r is said to be in *reduced normal form* if it has the form

$$\text{(rnf)}\qquad r=(\bigvee_{1\leqslant j\leqslant s}\phi_j)/p_0,$$

and each disjunct ϕ_j has the form

$$\phi_j=\bigwedge_{0\leqslant i\leqslant n}p_i^{t(i,j,0)}\wedge\bigwedge_{0\leqslant i\leqslant n}(\Diamond p_i)^{t(i,j,1)},$$

where (i) all ϕ_j are different, (ii) $p_0, \ldots, p_n$ denote propositional variables, (iii) t is a Boolean function $t : \{0, \ldots, n\} \times \{1, \ldots, s\} \times \{0, 1\} \to \{0, 1\}$ (i.e., $t(i, j, z) \in \{0, 1\}$), and (iv) $\alpha^0 \overset{\mathsf{def}}{=} \neg\alpha$ and $\alpha^1 \overset{\mathsf{def}}{=} \alpha$ for any formula α.

Using the renaming technique any modal rule can be transformed into an equivalent rule in reduced normal form [15].

Lemma 3.3 *Any rule $r = \alpha/\beta$ can be transformed in exponential time to an equivalent rule in reduced normal form.*

Proof We describe the algorithm of [18] (Lemma 3.1.3 and Theorem 3.1.11). Let $r = \alpha/\beta$ be a rule. We need a set of new variables $\{q_\gamma \mid \gamma \in \mathsf{sub}(\alpha, \beta)\}$. The first step is to replace $r = \alpha/\beta$ with $r_1 = \alpha \wedge (q_\beta \leftrightarrow \beta)/q_\beta$. It is easy to see that r is refuted on a frame F iff r_1 can be refuted on the same frame F. Therefore r and r_1 are equivalent.

Inductive step: Suppose the rule $r_i = \gamma_i/q_\beta$ was obtained in the ith step. We call a formula $\delta \in \mathsf{sub}(\gamma_i) \cap \mathsf{sub}(\alpha, \beta)$ *final*, if it is not a variable and not a proper subformula of any other formula in $\mathsf{sub}(\gamma_i) \cap \mathsf{sub}(\alpha, \beta)$. Let T_i be the set of all final formulae obtained at the ith step. We replace the rule r_i with a new one, namely $r_{i+1} = \gamma_{i+1}/q_\beta$, where

$$\gamma_{i+1} = t_i(\gamma_i) \wedge \bigwedge_{q_\gamma \vee q_\delta \in T_i} ((q_\gamma \leftrightarrow \gamma) \wedge (q_\delta \leftrightarrow \delta)) \wedge \bigwedge_{\neg q_\delta, \Diamond q_\delta \in T_i} (q_\delta \leftrightarrow \delta),$$

and $t_i(\gamma_i)$ is the formula obtained from γ_i by replacing all final subformulae δ with q_δ. It is straightforward to check that r_i and r_{i+1} are equivalent.

Note that every inductive step reduces the maximal height of the non-Boolean subformulae of the rule. Therefore after a finite number of steps we get a premise γ_k, which is a Boolean combination of *literals* of the form p or $\Diamond p$, where p is a propositional variable. We denote this intermediate form by $\mathsf{do}(r)$ (depth-one form).

Finally, we transform the premise of the obtained rule $r_N = \gamma_k/q_\beta$ into disjunctive normal form over literals. This requires no more than exponential time on the number of variables, i.e., on the number of subformulae of the original rule, which is the same as for the reduction of any Boolean formula to disjunctive normal form.□

The reduced normal form, obtained in Lemma 3.3, is uniquely defined and is denoted by $\mathsf{rnf}(r)$. Note that Lemma 3.3 proves more than the equivalence of r and $\mathsf{rnf}(r)$. In particular, from the proof it follows that if r is refutable in a model N then $\mathsf{rnf}(r)$ is refutable in a definable variant M of N with $M, w \models q_\gamma \iff N, w \models \gamma$ for all $\gamma \in \mathsf{sub}(\alpha, \beta)$.

Let r be any rule in reduced normal form (rnf). Let $\Theta(r) \overset{\mathsf{def}}{=} \{\phi_1, \ldots, \phi_s\}$ be the set of all disjuncts of r. For every $\phi_j \in \Theta(r)$, let

$$\theta(\phi_j) \overset{\mathsf{def}}{=} \{p_i \mid t(i, j, 0) = 1\} \quad \text{and} \quad \theta_\Diamond(\phi_j) \overset{\mathsf{def}}{=} \{p_i \mid t(i, j, 1) = 1\}.$$

That is, $\theta(\phi_j)$ denotes the set of variables of r with positive occurrences in ϕ_j, and $\theta_\Diamond(\phi_j)$ is the set of variables p_i of r with the positive occurrence of $\Diamond p_i$ in ϕ_j.

Historically the first algorithm for recognising admissible rules of $\mathsf{S4}$ was based on the next theorem. Its formulation requires the following definition, which is taken from [15]. For every subset of disjuncts $W \subseteq \Theta(\mathsf{rnf}(r))$, let $\mathcal{M}(\mathsf{rnf}(r), W)$ denote the Kripke model in which $W^M \stackrel{\mathsf{def}}{=} W$,

$$R^M \stackrel{\mathsf{def}}{=} \{(\phi_1, \phi_2) \mid \theta_\diamond(\phi_2) \subseteq \theta_\diamond(\phi_1)\} \quad \text{and} \quad (p_i, \phi_j) \in \nu^M \stackrel{\mathsf{def}}{\iff} p_i \in \theta(\phi_j).$$

Theorem 3.4 (Theorem 3.9.6 [18]) *A rule* $\mathsf{rnf}(r)$ *is admissible for* S4 *iff for any set* $W \subseteq \Theta(\mathsf{rnf}(r))$*, the model* $\mathcal{M}(\mathsf{rnf}(r), W)$ *fails to have at least one of the following properties.*

(a1) There is $\phi_j \in W$ *such that* $\mathcal{M}(\mathsf{rnf}(r), W), \phi_j \not\models p_0$.
(a2) $\mathcal{M}(\mathsf{rnf}(r), W), \phi_j \models \phi_j$ *for all* $\phi_j \in W$.
(a3) For any subset $\mathcal{D}$ *of* $\mathcal{M}$ *there exists* $\phi_j \in W$ *such that*

$$\theta_\diamond(\phi_j) = \theta(\phi_j) \cup \bigcup_{\phi \in \mathcal{D}} \theta_\diamond(\phi).$$

Note that in (a3), $\mathcal{D}$ can be empty.

Theorem 3.4 implies that we can employ this algorithm for recognising the admissibility of a rule r in S4 [15].

Step 1. Reduce rule r to depth-one-form $\mathsf{do}(r)$.
Step 2. Construct from the $\mathsf{do}(r)$ the reduced form $\mathsf{rnf}(r)$.
Step 3. For every subset W of the set of disjuncts of $\mathsf{rnf}(r)$, check whether the conditions of Theorem 3.4 hold.
Step 4. If the conditions (a1)–(a3) of Theorem 3.4 are satisfied for some W, then r is not admissible for S4, otherwise r is admissible for S4.

Step 1 can be done in polynomial time, Step 2 can be done in exponential time, and Step 3 can be done in exponential time. This means the time complexity of the algorithm is bounded by a doubly-exponential function in the length of r. A more detailed complexity analysis based on the techniques from [8,9] implies that the rule admissibility problem for S4 is coNExpTime-complete [10].

4 Semantic Characterisation of Admissibility

We now give a semantic characterisation of admissibility in S4. We say that an $\mathsf{S4^u}$-model satisfies the *formula definable co-cover property* if the following (infinite) set of axioms hold (for $n > 0$):

$$\text{(FCCP)} \qquad \exists x \forall p \, (\nu(\diamond p, x) \to \nu(p, x))$$
$$\forall x_1 \cdots \forall x_n \exists x \forall p \, \big(R(x, x_1) \wedge \cdots \wedge R(x, x_n) \wedge \\ \nu(\diamond p, x) \to (\nu(p, x) \vee \nu(\diamond p, x_1) \vee \cdots \vee \nu(\diamond p, x_n))\big).$$

Let $S_{\mathcal{FCCP}}$ be the semantic specification consisting of the (FCCP) formulae and the formulae in $S_{\mathsf{S4^u}}$. Let $\mathcal{FCCP}(\Sigma)$ be the class of all $\mathsf{S4^u}$-models of the signature Σ that satisfy all instances of the formulae of $S_{\mathcal{FCCP}}$ under substitutions of $\mathcal{L}^{\mathsf{u}}_{\mathsf{S4}}$-formulae in Σ for propositional variables.

Let $\mathsf{S4^{u,+}}$ be the modal logic with the language $\mathcal{L}^{\mathsf{u}}_{\mathsf{S4}}$ that has $S_{\mathcal{FCCP}}$ as semantic specification. The following theorem is a direct consequence of the definitions above and the fact that all n-characterising models $\mathsf{Ch}_n(\mathsf{S4})$ for $\mathsf{S4}$ satisfy the (FCCP) formulae.

Theorem 4.1 *$\mathsf{S4}^{\mathsf{u},+}$ is a conservative extension of $\mathsf{S4}$.*

Now we prove that $\mathsf{S4}^{\mathsf{u},+}$ has the effective finite model property. Let Σ be a *fixed* signature and M an $\mathsf{S4}^{\mathsf{u},+}$-model of the signature Σ. We define the *filtrated* (through Σ) model $\overline{M}$ as follows. The equivalence $\sim$ on the set W^M is defined by $w \sim v \overset{\mathsf{def}}{\iff} \tau(w) = \tau(v)$. Further, $[w] \overset{\mathsf{def}}{=} \{v \in W^M \mid w \sim v\}$ and $W^{\overline{M}} \overset{\mathsf{def}}{=} \{[w] \mid w \in W^M\}$. Next, $R^{\overline{M}} \overset{\mathsf{def}}{=} \{([w],[v]) \mid M, v \models \psi$ implies $M, w \models \Diamond\psi$ for every $\Diamond\psi \in \Sigma\}$ and, finally, $\nu^{\overline{M}} \overset{\mathsf{def}}{=} \{(\psi,[w]) \mid (\psi, w) \in \nu^M\}$ for every $\psi \in \Sigma$.

The following lemma can be proved by induction on the structure of formulae in Σ by verifying that all semantic conditions in $S_{\mathcal{FCCP}}$ hold.

Lemma 4.2 (Filtration Lemma) *$\overline{M}$ is an $\mathsf{S4}^{\mathsf{u},+}$-model of the signature Σ.*

Note that by definition $\overline{M}$ is Σ-differentiated and it is finite whenever Σ is finite.

Theorem 4.3 (The Effective Finite Model Property) *Let ϕ be a formula and n the length of ϕ (i.e., the number of symbols in ϕ). If ϕ is satisfiable in an $\mathsf{S4}^{\mathsf{u},+}$-model (of the signature $\mathsf{sub}(\phi)$) then it is satisfiable in a finite $\mathsf{S4}^{\mathsf{u},+}$-model (of the signature $\mathsf{sub}(\phi)$) and its size does not exceed 2^n.*

Theorem 4.4 *$\alpha/\beta \in \mathsf{Adm}(\mathsf{S4})$ iff $[\mathsf{u}]\alpha \wedge \neg\beta$ is unsatisfiable in $\mathsf{S4}^{\mathsf{u},+}$.*

Proof Suppose α/β is not admissible and $p_1, \ldots, p_n$ are all the propositional variables occurring in α and β. Then there is a model M—a definable variant of $\mathsf{Ch}_n(\mathsf{S4})$ such that $M \not\models \alpha/\beta$. This model has the co-cover property and it is routine to transform M into an $\mathsf{S4}^{\mathsf{u},+}$-model satisfying $[\mathsf{u}]\alpha \wedge \neg\beta$.

For the converse, let $\Sigma \overset{\mathsf{def}}{=} \mathsf{sub}([\mathsf{u}]\alpha \wedge \neg\beta)$ and suppose $[\mathsf{u}]\alpha \wedge \neg\beta$ is satisfiable in an $\mathsf{S4}^{\mathsf{u},+}$-model. Then it is satisfiable in a finite Σ-differentiated $\mathsf{S4}^{\mathsf{u},+}$-model M (of the signature Σ), by the Filtration Lemma. Let $\mathsf{sub}_\Diamond(\alpha,\beta) \overset{\mathsf{def}}{=} \{\Diamond\gamma \mid \gamma \in \mathsf{sub}(\alpha,\beta)\} \cup \mathsf{sub}(\alpha,\beta)$ for any $\mathcal{L}_{\mathsf{S4}}$-formulae α and β. For every $w \in W^M$ let $\tau_\Diamond(w) \overset{\mathsf{def}}{=} \{\gamma \in \mathsf{sub}_\Diamond(\alpha,\beta) \mid M, w \models \gamma\}$ and

$$\phi(w) \overset{\mathsf{def}}{=} \bigwedge_{\gamma \in \mathsf{sub}(\alpha,\beta)} q_\gamma^{\chi(\gamma)} \wedge \bigwedge_{\gamma \in \mathsf{sub}(\alpha,\beta)} \Diamond q_\gamma^{\chi(\Diamond\gamma)},$$

where χ is the characteristic function of the set $\tau_\Diamond(w)$. Let us consider the model $M^* \overset{\mathsf{def}}{=} \langle W^{M^*}, R^{M^*}, \nu^{M^*}\rangle$, where $W^{M^*} \overset{\mathsf{def}}{=} \{\phi(w) \mid w \in W^M\}$, $(\phi(u),\phi(v)) \in R^{M^*} \overset{\mathsf{def}}{\iff} (u,v) \in R^M$, and $(q_\gamma, \phi(w)) \in \nu^{M^*} \overset{\mathsf{def}}{\iff} M, w \models \gamma$. Each $\phi(w)$ is a disjunct in reduced normal form $\mathsf{rnf}(r)$. Therefore we have that

- $W^{M^*} \subseteq \Theta(\mathsf{rnf}(r))$,
- $\langle W^{M^*}, R^{M^*}\rangle$ is isomorphic to the underlying frame of M,
- M^* satisfies conditions (a1)–(a3) of Theorem 3.4.

By Theorem 3.4, $\mathsf{rnf}(r)$ is not admissible, and hence neither is r. □

5 Synthesising a Tableau Calculus

We now apply the method of [22] to generate a sound and constructively complete tableau calculus for $\mathsf{S4}^{\mathsf{u},+}$. In order to apply the method we must ensure that the semantic specification $S_{\mathcal{FCCP}}$ of $\mathsf{S4}^{\mathsf{u},+}$ is well-defined in the sense of [22]. First, $S_{\mathcal{FCCP}}$ must be normalised in the following sense: (i) all the formulae specifying semantics are $\mathcal{L}^{\mathsf{u}}_{\mathsf{S4}}$-open, i.e., they do not contain quantifiers of propositional variables, and (ii) all the formulae in $S_{\mathcal{FCCP}}$ are divided into three groups: positive and negative definitions of the semantics of the connectives of the logic, and a background theory that imposes frame conditions. It is also required that all formulae in the background theory do not contain any non-atomic modal terms.

Every definition of the $\mathcal{L}^{\mathsf{u}}_{\mathsf{S4}}$ connectives in $S_{\mathcal{FCCP}}$ (in Figure 1 and (1)) can be split into two implications. The resulting set of formulae can be divided into the required two groups of positive and negative connective definitions. The third group, the background theory of $\mathsf{S4}^{\mathsf{u},+}$, consists of formulae specifying the reflexivity and transitivity for the relation R and the (FCCP) formulae.

The main difficulties for the normalisation of the specification $S_{\mathcal{FCCP}}$ are the occurrences of the non-atomic modal term $\Diamond p$ and the quantification of the variable p in (FCCP). To solve this problem we first replace every formula $\nu(\Diamond p, y)$ by its semantic equivalent $\exists z(R(y,z) \wedge \nu(p,z))$ and transform the resulting formulae into the prenex normal form. This gives us:

$$\exists x \forall p \forall y \big((R(x,y) \wedge \nu(p,y)) \rightarrow \nu(p,x)\big)$$

$$\begin{aligned}&\forall x_1 \cdots \forall x_n \exists x \forall p \forall y \exists z \,\big(R(x,x_1) \wedge \cdots \wedge R(x,x_n) \wedge \\ &\quad (R(x,y) \wedge \nu(p,y)) \rightarrow \big(\nu(p,x) \vee (\nu(p,z) \wedge (R(x_1,z) \vee \cdots \vee R(x_n,z)))\big)\big).\end{aligned}$$

These formulae are still not $\mathcal{L}^{\mathsf{u}}_{\mathsf{S4}}$-open formulae as required in [22] because of the quantifiers on p. However, using Skolemisation it is possible to eliminate all existential quantifiers preceding the p-quantifiers in the formulae and then we can omit the quantifiers of p. In addition, we split the long formulae in two parts. We get:

(FCCP′)
$$\forall y \big((R(g_0,y) \wedge \nu(p,y)) \rightarrow \nu(p,g_0)\big)$$

$$\forall x_1 \cdots \forall x_n \big(R(g_n(x_1,\ldots,x_n),x_1) \wedge \cdots \wedge R(g_n(x_1,\ldots,x_n),x_n)\big)$$

$$\begin{aligned}&\forall x_1 \cdots \forall x_n \forall y \exists z \,\big((R(g_n(x_1,\ldots,x_n),y) \wedge \nu(p,y)) \rightarrow \\ &\quad \big(\nu(p,g_n(x_1,\ldots,x_n)) \vee (\nu(p,z) \wedge (R(x_1,z) \vee \cdots \vee R(x_n,z)))\big)\big).\end{aligned}$$

Here, g_n $(n \geq 0)$ denote the introduced Skolem symbols.

We use the notation $S'_{\mathcal{FCCP}}$ for the semantic specification obtained from $S_{\mathcal{FCCP}}$ where all (FCCP) formulae have been replaced by the corresponding (FCCP′) formulae. It is clear that for every first-order structure M, the universal closure of $S_{\mathcal{FCCP}}$ and the universal closure of $S'_{\mathcal{FCCP}}$ are equisatisfiable in M. Hence, the transformed semantics $S'_{\mathcal{FCCP}}$ is equivalent to $S_{\mathcal{FCCP}}$ and axiomatises the same class of models.

Now we are ready to synthesise tableau calculi from $S'_{\mathcal{FCCP}}$. The generated tableau rules are given in Figure 2. The symbols $f_{\Diamond}$, f_{u}, g_n, h_n denote Skolem functions and g_0 denotes a Skolem constant.

Decomposition tableau rules:

$$\frac{\nu(\neg p, x)}{\neg\nu(p, x)} \qquad \frac{\neg\nu(\neg p, x)}{\nu(p, x)}$$

$$\frac{\nu(p \vee q, x)}{\nu(p, x) \mid \nu(q, x)} \qquad \frac{\neg\nu(p \vee q, x)}{\neg\nu(p, x),\ \neg\nu(q, x)}$$

$$\frac{\nu(\Diamond p, x)}{R(x, f_\Diamond(p, x)),\ \nu(p, f_\Diamond(p, x))} \qquad \frac{\nu(\neg\Diamond p, x)}{\neg R(x, y) \mid \neg\nu(p, y)}$$

$$\frac{\nu(\langle \mathsf{u} \rangle p, x)}{\nu(p, f_{\mathsf{u}}(p))} \qquad \frac{\nu(\neg\langle \mathsf{u} \rangle p, x),\ y \approx y}{\neg\nu(p, y)}$$

Theory tableau rules:

$$\frac{x \approx x}{R(x, x)} \qquad \frac{x \approx x,\ y \approx y,\ z \approx z}{\neg R(x, y) \mid \neg R(y, z) \mid R(x, z)}$$

Infinite set of (FCCP) tableau rules ($n > 0$):

$$(\mathsf{cc}^0)\text{:}\ \frac{p \approx p,\ y \approx y}{\neg R(g_0, y) \mid \neg\nu(p, y) \mid \nu(p, g_0)} \qquad (\mathsf{cc}_0^n)\text{:}\ \frac{x_1 \approx x_1,\ \ldots,\ x_n \approx x_n,\ y \approx y}{R(g_n(\overline{x}), x_1),\ \ldots,\ R(g_n(\overline{x}), x_n)}$$

$$(\mathsf{cc}_1^n)\text{:}\ \frac{p \approx p,\ x_1 \approx x_1,\ \ldots,\ x_n \approx x_n,\ y \approx y}{\neg R(g_n(\overline{x}), y) \mid \neg\nu(p, y) \mid \nu(p, g_n(\overline{x})) \mid R(x_1, h_n(p, \overline{x}, y)),\ \nu(p, h_n(p, \overline{x}, y)) \mid \cdots}$$

Closure tableau rules:

$$\frac{\nu_1(p, x),\ \neg\nu_1(p, x)}{\bot} \qquad \frac{R(x, y),\ \neg R(x, y)}{\bot}$$

Figure 2. Generated tableau rules for $\mathsf{S4}^{\mathsf{u},+}$

$$\frac{R(x, y)}{x \approx x,\ y \approx y} \qquad \frac{\neg R(x, y)}{x \approx x,\ y \approx y} \qquad \frac{\neg\nu(p, x)}{p \approx p,\ x \approx x} \qquad \frac{\nu(p, x)}{p \approx p,\ x \approx x}$$

$$\frac{R(x, y),\ x \approx z}{R(z, y)} \qquad \frac{R(x, y),\ y \approx z}{R(x, z)} \qquad \frac{\nu(p, x),\ x \approx y}{\nu(p, y)} \qquad \frac{x \approx y}{y \approx x} \qquad \frac{x \approx y,\ y \approx z}{x \approx z}$$

Figure 3. Equality congruence rules for predicates occurring in $S_{\mathsf{S4}^{\mathsf{u}}}$.

Let T be the tableau calculus consisting of the rules of Figure 2 and the standard tableau rules for equality listed in Figure 3. The equality tableau rules are obtained from the equality congruence axioms, which are always included in the background theory of any semantic specification, see [22].

Given a formula ϕ, and assuming our aim is to determine the satisfiability of ϕ, the start of any tableau derivation is the formula $\nu(\phi, a)$, where a is an arbitrary constant a that does not occur in the rules of T. a can be viewed as a Skolem constant introduced for $\exists x$ in the formula $\exists x\, \nu(\phi, x)$. The rules of the calculus are applied top-down in the familiar way.

We assume the following definitions from [21,22]. Let T denote a tableau calculus and ϕ is a formula. We denote by $T(\phi)$ a finished tableau derivation for testing the satisfiability of ϕ. That is, all branches in the tableau derivation are fully expanded and all applicable rules of T have been applied in $T(\phi)$. As usual we assume that all the rules of the calculus are applied non-deterministically. A branch of a tableau derivation is *closed* if a closure rule has been applied, otherwise the branch is called *open*. The tableau derivation $T(\phi)$ is *closed* if all its branches are closed and $T(\phi)$ is *open* otherwise. A calculus T is *sound* (for a logic L) iff for any formula ϕ, each $T(\phi)$ is open whenever ϕ is satisfiable in an L-model. T is *complete* iff for any unsatisfiable formula ϕ there is a $T(\phi)$ which is closed. T is *constructively complete* (for L) iff for any open branch in a finished tableau derivation in T it is possible to construct an L-model from terms of the branch such that the model reflects all the formulae occurring in the branch. (Constructive completeness is a stronger notion than completeness.) T is said to be *terminating* if every finished open tableau derivation in T has a finite open branch.

It is easy to check that the specification $S'_{\mathcal{FCCP}}$ for $\mathsf{S4}^{\mathsf{u},+}$ is well-defined in the sense of [22]. A consequence of [22, Theorems 1 and 2] is this result.

Theorem 5.1 *T is a sound and constructively complete calculus for $\mathsf{S4}^{\mathsf{u},+}$.*

6 A Refined Tableau Calculus

The generated calculus T can be refined as follows. First, it is possible to refine the calculus by moving negated conclusions in certain rules up to premise positions. We move formulae that contain only propositional variables and do not contain any complex modal terms upwards. It is not difficult to check for each rule that the condition given in [22, Theorem 3] for this refinement to preserve soundness and constructive completeness is true.

Second, we can apply the second refinement described in [22, Section 5]. For this we extend the language $\mathcal{L}^{\mathsf{u}}_{\mathsf{S4}}$ by introducing a countable set $\{i, j, k, \ldots\}$ of nominal variables and logical connectives (acting on nominals) which correspond to Skolem functions and constants. The @ operator can be introduced and specified in such a way that $@_i\phi \overset{\mathsf{def}}{=} [\mathsf{u}](i \to \phi)$ for every nominal term i and formula ϕ (of the extended language). It is not difficult to see that the semantics of all the connectives of the extended language can be represented in the language itself (see [22, Section 5]).

Summing up, the refined rules we obtain are given in Figure 4. In these rules, $i, j, k, i_1, \ldots, i_n$ denote nominal variables, and $f_\Diamond$, f_{u}, g_n, and h_n ($n \geq 0$) denote 'nominal functions' which correspond to the Skolem functions with same names.

Let T^+ be the tableau calculus which consists of the rules of Figure 4 and the refined equality rules given in Figure 5. In T^+, a tableau derivation for testing the satisfiability of ϕ starts with a formula $@_{i_0}\phi$ where i_0 is a *fresh* nominal constant.

Theorem 6.1 *T^+ is a sound and constructively complete tableau calculus for $\mathsf{S4}^{\mathsf{u},+}$.*

7 A Terminating Tableau Calculus

Our proof of Theorem 4.3 that $\mathsf{S4}^{\mathsf{u},+}$ has the effective finite model property uses a filtration argument. That is, in the terminology of [21], $\mathsf{S4}^{\mathsf{u},+}$ admits finite filtration. Using the results of [21], this means that the tableau calculi generated in Section 5 and 6 can be turned into terminating tableau calculi. In particular, we are interested only in the refined calculus T^+.

Adding the following unrestricted blocking rule to T^+ gives a terminating tableau calculus.

$$(\mathsf{ub}):\ \frac{@_i i,\ @_j j}{@_i j \mid @_i \neg j}$$

The conditions that blocking must satisfy are:

Decomposition tableau rules:

$$(\neg)\colon \frac{@_i\neg\neg p}{@_i p}$$

$$(\vee)\colon \frac{@_i(p\vee q)}{@_i p \mid @_i q} \qquad (\neg\vee)\colon \frac{@_i\neg(p\vee q)}{@_i\neg p,\ @_i\neg q}$$

$$(\Diamond)\colon \frac{@_i\Diamond p}{@_i\Diamond f_\Diamond(p,i),\ @_{f_\Diamond(p,i)}p} \qquad (\neg\Diamond)\colon \frac{@_i\neg\Diamond p,\ @_i\Diamond j}{@_j\neg p}$$

$$(\langle\mathsf{u}\rangle)\colon \frac{@_i\langle\mathsf{u}\rangle p}{@_{f_{\mathsf{u}}(p)}p} \qquad (\neg\langle\mathsf{u}\rangle)\colon \frac{@_i\neg\langle\mathsf{u}\rangle p,\ @_j j}{@_j\neg p}$$

Theory tableau rules:

$$(\mathsf{refl})\colon \frac{@_i i}{@_i\Diamond i} \qquad (\mathsf{trans})\colon \frac{@_i\Diamond j,\ @_j\Diamond k}{@_i\Diamond k}$$

Infinite set of (FCCP′) tableau rules ($n > 0$):

$$(\mathsf{cc}'^0_0)\colon \frac{}{@_{g_0}g_0} \qquad (\mathsf{cc}'^0_1)\colon \frac{@_{g_0}\Diamond i,\ @_i p}{@_{g_0}p}$$

$$(\mathsf{cc}'^n_0)\colon \frac{@_{i_1}i_1,\ \ldots,\ @_{i_n}i_n}{@_{g_n(\bar{i})}\Diamond i_1,\ \ldots,\ @_{g_n(\bar{i})}\Diamond i_n}$$

$$(\mathsf{cc}'^n_1)\colon \frac{@_{g_n(\bar{i})}\Diamond j,\ @_j p}{@_{g_n(\bar{i})}p \mid @_{i_1}\Diamond h_n(p,\bar{i},j),\ @_{h_n(p,\bar{i},j)}p \mid \cdots \mid @_{i_n}\Diamond h_n(p,\bar{i},j),\ @_{h_n(p,\bar{i},j)}p}$$

Closure tableau rules:

$$(\bot)\colon \frac{@_i p,\ @_i\neg p}{\bot}$$

Figure 4. Refined tableau rules for $\mathsf{S4}^{\mathsf{u},+}$

$$(\mathsf{refl}{\approx})\colon \frac{@_i p}{@_i i} \qquad (\mathsf{sym}{\approx})\colon \frac{@_i j}{@_j i} \qquad (\mathsf{trans}{\approx})\colon \frac{@_i j,\ @_j k}{@_i k}$$

$$(\mathsf{con}{\approx}_0)\colon \frac{@_i p,\ @_i j}{@_j p} \qquad (\mathsf{con}{\approx}_1)\colon \frac{@_i\Diamond j,\ @_j k}{@_i\Diamond k}$$

Figure 5. Refined equality congruence rules

(b1) The rules ($\Diamond$) and ($\langle\mathsf{u}\rangle$) are never applied to formulae of the form $@_i\Diamond j$ and, respectively, $@_i\langle\mathsf{u}\rangle j$ where j is a nominal term.

(b2) If $@_i j$ appears in a branch and $i < j$ (i.e., nominal i appeared strictly earlier than nominal j in the derivation) then all further applications of the tableau rules which produce new nominals (in our case the ($\Diamond$), ($\langle\mathsf{u}\rangle$), (cc'^0_0) and (cc'^n_0) rules) to the formulae with occurrences of j are not performed within the branch.

(b3) In every open branch there is some node from which point onwards before any application of any tableau rule that produces new nominals (i.e., the ($\Diamond$), ($\langle\mathsf{u}\rangle$), (cc'^0_0) and (cc'^n_0) rules) all possible applications of the (ub) rule have been performed.

We denote the extended calculus by $T_{\mathsf{S4}^{\mathsf{u},+}}$.

Since T^+ is sound and constructively complete for $\mathsf{S4}^{\mathsf{u},+}$, and $\mathsf{S4}^{\mathsf{u},+}$ admits finite filtration the results in [20] allow us to state:

Theorem 7.1 *$T_{\mathsf{S4}^{\mathsf{u},+}}$ is a sound, (constructively) complete and terminating tableau* *calculus for* $\mathsf{S4}^{u,+}$.

Let $T_{\mathsf{S4}^{\mathsf{u}}}$ be the tableau calculus which consists of the same set of rules as $T_{\mathsf{S4}^{\mathsf{u},+}}$ but excludes the (FCCP′) rules: (cc'^{0}_{0}), (cc'^{0}_{1}), (cc'^{n}_{0}), and (cc'^{n}_{1}). Applying tableau synthesis to $\mathsf{S4}^{\mathsf{u}}$ in a similar way gives the following result.

Theorem 7.2 *$T_{\mathsf{S4}^{u}}$ is a sound, (constructively) complete and terminating tableau calculus for* $\mathsf{S4}^{u}$.

8 A Tableau Method for Testing Rule Admissibility

Putting all the results together (in particular Theorems 3.1, 4.4, 7.1 and 7.2) here is a method for determining whether a modal rule is admissible in $\mathsf{S4}$, or not.

Step 1. Given an $\mathsf{S4}$-rule α/β, rewrite it to $[\mathsf{u}]\alpha \wedge \neg\beta$.

Step 2. Use the tableau calculus $T_{\mathsf{S4}^{\mathsf{u}}}$ to test the satisfiability of $[\mathsf{u}]\alpha \wedge \neg\beta$ in $\mathsf{S4}^{\mathsf{u}}$.

Step 3. If $T_{\mathsf{S4}^{\mathsf{u}}}([\mathsf{u}]\alpha \wedge \neg\beta)$ is closed, i.e., $[\mathsf{u}]\alpha \wedge \neg\beta$ is unsatisfiable in $\mathsf{S4}^{\mathsf{u}}$, then stop and return **'derivable'**.

Step 4. Otherwise (i.e., $T_{\mathsf{S4}^{\mathsf{u}}}([\mathsf{u}]\alpha \wedge \neg\beta)$ is open), *continue* the tableau derivation with the rules in $T_{\mathsf{S4}^{\mathsf{u}}}$ plus the rules (cc'^{0}_{0}), (cc'^{0}_{1}), (cc'^{n}_{0}), and (cc'^{n}_{1}). In particular, continue the derivation with a finite open branch of $T_{\mathsf{S4}^{\mathsf{u}}}([\mathsf{u}]\alpha \wedge \neg\beta)$ using the rules of $T_{\mathsf{S4}^{\mathsf{u},+}}$ until the derivation stops.

Step 5. If $T_{\mathsf{S4}^{\mathsf{u},+}}([\mathsf{u}]\alpha \wedge \neg\beta)$ is closed then return **'not derivable and admissible'**. Otherwise, i.e., if $T_{\mathsf{S4}^{\mathsf{u},+}}([\mathsf{u}]\alpha \wedge \neg\beta)$ is open, return **'not admissible'**.

The answers returned by the method are either **'derivable'**, **'not derivable and admissible'**, or **'not admissible'**. If a rule is derivable it is also admissible but not conversely.

Figure 6 demonstrates the algorithm for the rule $\Diamond p \wedge \Diamond\neg p/\bot$. The Step 1 rewrites the rule to $\neg\langle\mathsf{u}\rangle(\neg\Diamond p \vee \neg\Diamond\neg p)$ modulo standard modal equivalences. The black triangles in the figure denote branching points in the derivation. A branch expansion after a branching point is indicated by appropriate indentation.

9 Concluding Remarks

A major difficulty in dealing with $\mathsf{S4}$-admissibility, is that the theory of $\mathsf{S4}$-admissible rules does not have the finite approximation property [11] in this sense: for a rule $r \notin \mathsf{Adm}(\mathsf{S4})$, there is no single finite Kripke model M *separating* r from $\mathsf{Adm}(\mathsf{S4})$, (i.e., such that $M \models \mathsf{Adm}(\mathsf{S4})$, but $M \not\models r$). Therefore the tableau algorithm that we have introduced in this paper builds open branches that represent a (possibly) infinite $\mathsf{Adm}(\mathsf{S4})$-model and is a counter-model to a non-admissible rule. The subsequent filtration provides a finite model (not necessarily an $\mathsf{Adm}(\mathsf{S4})$-model) witnessing the refutation.

A known algorithm that can handle $\mathsf{S4}$-admissibility appears in Zucchelli [24] and is based on the research of Ghilardi [3,4,5,6] on projective approximations. This algorithm is based on the algorithm from [5] for IPC, which combines resolution and tableau approaches for finding projective approximations of a formula and relies on the existence of an algorithm for theorem proving. This algorithm (as well as

1.	$@_{i_0}\neg\langle\mathsf{u}\rangle(\neg\Diamond p\vee\neg\Diamond\neg p)$	given
2.	$@_{i_0} i_0$	$(\mathrm{refl}\approx)$,1
3.	$@_{i_0}\neg(\neg\Diamond p\vee\neg\Diamond\neg p)$	$(\neg\langle\mathsf{u}\rangle)$,1,2
4.	$@_{i_0}\neg\neg\Diamond p$	$(\neg\vee)$,3
5.	$@_{i_0}\neg\neg\Diamond\neg p$	$(\neg\vee)$,3
6.	$@_{i_0}\Diamond p$	$(\neg)$,4
7.	$@_{i_0}\Diamond\neg p$	$(\neg)$,5
8.	$@_{i_0}\Diamond i_1$	$(\Diamond)$,6: $i_1 \overset{\mathrm{def}}{=} f_\Diamond(p,i_0)$
9.	$@_{i_1} p$	$(\Diamond)$,6
10.	$@_{i_1} i_1$	$(\mathrm{refl}\approx)$,9
11.	$@_{i_0}\Diamond i_2$	$(\Diamond)$,7: $i_2 \overset{\mathrm{def}}{=} f_\Diamond(\neg p,i_0)$
12.	$@_{i_2}\neg p$	$(\Diamond)$,7
13.	$@_{i_2} i_2$	$(\mathrm{refl}\approx)$,12
14.	$@_{i_2}\neg(\neg\Diamond p\vee\neg\Diamond\neg p)$	$(\neg\langle\mathsf{u}\rangle)$,1,13
15.	$@_{i_2}\neg\neg\Diamond p$	$(\neg\vee)$,14
16.	$@_{i_2}\neg\neg\Diamond\neg p$	$(\neg\vee)$,14
17.	$@_{i_2}\Diamond p$	$(\neg)$,15
18.	$@_{i_2}\Diamond\neg p$	$(\neg)$,16
19.	▶$@_{i_0} i_1$	(ub),2,10
20.	$@_{i_1} i_0$	$(\mathrm{sym}\approx)$,19
21.	$@_{i_0} p$	$(\mathrm{con}\approx_0)$,9,20
22.	$@_{i_1}\Diamond i_1$	$(\mathrm{con}\approx_0)$,8,19
23.	$@_{i_0}\Diamond i_0$	$(\mathrm{con}\approx_1)$,8,20
24.	▶$@_{i_0} i_2$	(ub),2,13
25.	$@_{i_2} i_0$	$(\mathrm{sym}\approx)$,24
26.	$@_{i_0}\neg p$	$(\mathrm{con}\approx_0)$,12,25
	Unsatisfiable	21,26
27.	▶$@_{i_0}\neg i_2$	(ub),2,13
28.	$@_{i_2}\Diamond i_3$	$(\Diamond)$,17: $i_3 \overset{\mathrm{def}}{=} f_\Diamond(p,i_2)$
29.	$@_{i_3} p$	$(\Diamond)$,17
30.	$@_{i_3} i_3$	$(\mathrm{refl}\approx)$,29
31.	$@_{i_2}\Diamond i_4$	$(\Diamond)$,18: $i_4 \overset{\mathrm{def}}{=} f_\Diamond(\neg p,i_2)$
32.	$@_{i_4}\neg p$	$(\Diamond)$,18
33.	$@_{i_4} i_4$	$(\mathrm{refl}\approx)$,32
34.	▶$@_{i_0} i_3$	(ub),2,30
35.	$@_{i_3} i_0$	$(\mathrm{sym}\approx)$,34
36.	$@_{i_2}\Diamond i_0$	$(\mathrm{con}\approx_1)$,28,35
37.	$@_{i_2}\Diamond i_2$	(trans),11,36
38.	▶$@_{i_2} i_4$	(ub),13,33
. . .		
	Satisfiable in S4$^{\mathsf{u}}$	
	The rule is **not derivable**	
39.	$@_{g_0} g_0$	(cc'^{0}_{0})
40.	$@_{g_0}\neg(\neg\Diamond p\vee\neg\Diamond\neg p)$	$(\neg\langle\mathsf{u}\rangle)$,1,35
41.	$@_{g_0}\neg\neg\Diamond p$	$(\neg\vee)$,40
42.	$@_{g_0}\neg\neg\Diamond\neg p$	$(\neg\vee)$,40
43.	$@_{g_0}\Diamond p$	$(\neg)$,41
44.	$@_{g_0}\Diamond\neg p$	$(\neg)$,42
45.	$@_{g_0}\Diamond i_5$	$(\Diamond)$,43: $i_5 \overset{\mathrm{def}}{=} f_\Diamond(p,g_0)$
46.	$@_{i_5} p$	$(\Diamond)$,43
47.	$@_{g_0}\Diamond i_6$	$(\Diamond)$,44: $i_6 \overset{\mathrm{def}}{=} f_\Diamond(\neg p,g_0)$
48.	$@_{i_6}\neg p$	$(\Diamond)$,44
49.	$@_{g_0} p$	(cc'^{0}_{1}),45,46
50.	$@_{g_0}\neg p$	(cc'^{0}_{1}),47,48
	Unsatisfiable in S4$^{\mathsf{u},+}$	49,50
51.	▶$@_{i_2}\neg i_4$	(ub),13,33
. . .		Similarly to 39–50
	Unsatisfiable in S4$^{\mathsf{u},+}$	
52.	▶$@_{i_0}\neg i_3$	(ub),2,30
. . .		Similarly to 39–50
	Unsatisfiable in S4$^{\mathsf{u},+}$	
53.	▶$@_{i_0}\neg i_1$	(ub),2,10
. . .		Similarly to 39–50
	Unsatisfiable in S4$^{\mathsf{u},+}$	
	The rule is **not derivable and admissible**	

Figure 6. A derivation for testing admissibility of $\Diamond p\wedge\Diamond\neg p/\bot$.

its precursor for IPC [5]) is specifically constructed for describing general solutions (maximal general unifiers, maximal since there can be more than one) for matching and unification problems (all other solutions can be obtained as substitution variants of general solutions). Applicability of this algorithm to the admissibility problem is a side effect, depending on some specific properties of S4 (see [6]). In contrast, the original algorithm of [15] can be used to find only *some* solution of matching and unification problems in S4. In particular, this can be done through the relation:

$$\text{an equation } \alpha\equiv\beta \text{ is solvable} \quad\text{ iff }\quad \text{a rule } \alpha\equiv\beta/\bot \text{ is not admissible.}$$

We expect that our method can be modified for finding the general solutions of logical equations as well.

Recently Wolter and Zakharyaschev [23] showed that modal logics with the *universal modality* that are situated between K$^{\mathsf{u}}$ and K4$^{\mathsf{u}}$ are undecidable with respect to admissibility and even with respect to only unification. They posed the question [23] whether the logic S4$^{\mathsf{u}}$ is decidable with respect to admissibility. This question is solved positively in Rybakov [19]. We strongly believe that our tableau method can also be extended to rule admissibility in S4$^{\mathsf{u}}$.

In fact, our method of replacing the first-order co-cover condition with its *formulaic* variant (which is also first-order but in the extended language) is not restricted to S4. We expect that it can be modified to deal with a number of other modal logics, especially those covered by the general theory of [18]. Similarly, it can be applied to their superintuitionistic counterparts (either through Gödel's translation or directly) and to transitive modal logics augmented with the universal modality [19].

References

[1] S. Babenyshev. The decidability of admissibility problems for modal logics S4.2 and S4.2Grz and superintuitionistic logic KC. *Algebra Logic*, 31(4):205–216, 1992.

[2] H. Friedman. One hundred and two problems in mathematical logic. *J. Symb. Log.*, 40(3):113–130, 1975.

[3] S. Ghilardi. Unification in intuitionistic logic. *J. Symb. Log.*, 64(2):859–880, 1999.

[4] S. Ghilardi. Best solving modal equations. *Ann. Pure Appl. Logic*, 102(3):183–198, 2000.

[5] S. Ghilardi. A resolution/tableaux algorithm for projective approximations in IPC. *Log. J. IGPL*, 10(3):229–243, 2002.

[6] S. Ghilardi and L. Sacchetti. Filtering unification and most general unifiers in modal logic. *J. Symb. Log.*, 69(3):879–906, 2004.

[7] R. Harrop. Concerning formulas of the types $a \to b \vee c$, $a \to \exists x b(x)$ in intuitionistic formal system. *J. Symb. Log.*, 25:27–32, 1960.

[8] R. Iemhoff. On the admissible rules of intuitionistic propositional logic. *J. Symb. Log.*, 66(1):281–294, 2001.

[9] E. Jeřábek. Admissible rules of modal logics. *J. Log. Comput.*, 15(4):411–431, 2005.

[10] E. Jeřábek. Complexity of admissible rules. *Arch. Math. Logic*, 46(2):73–92, 2007.

[11] V. R. Kiyatkin, V. V. Rybakov, and T. Oner. On finite model property for admissible rules. *Mathematical Logic Quarterly*, 45:505–520, 1999.

[12] P. Lorenzen. *Einführung in die operative Logik und Mathematik.* Springer, 1955.

[13] G. Mints. Derivability of admissible rules. *Journal of Soviet Mathematics*, 6(4):417–421, 1976.

[14] P. Roziere. Admissible and derivable rules. *Mathematical Structures in Computer Science*, (3):129–136, 1993.

[15] V. V. Rybakov. A criterion for admissibility of rules in modal system S4 and the intuitionistic logic. *Algebra Logic*, 23(5):369–384, 1984.

[16] V. V. Rybakov. Semantic admissibility criteria for deduction rules in S4 and Int. *Mat. Zametki*, 50(1):84–91, 1991.

[17] V. V. Rybakov. Rules of inference with parameters for intuitionistic logic. *J. Symb. Log.*, 57(3):912–923, 1992.

[18] V. V. Rybakov. *Admissibility of logical inference rules*, vol. 136 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 1997.

[19] V. V. Rybakov. Logics with universal modality and admissible consecutions. *J. Appl. Non-Classical Log.*, 17(3):381–394, 2007.

[20] R. A. Schmidt and D. Tishkovsky. Using tableau to decide expressive description logics with role negation. In *ISWC07*, vol. 4825 of *Lect. Notes Comput. Sci.*, pp. 438–451. Springer, 2007.

[21] R. A. Schmidt and D. Tishkovsky. A general tableau method for deciding description logics, modal logics and related first-order fragments. In *IJCAR08*, vol. 5195 of *Lect. Notes Comput. Sci.*, pp. 194–209. Springer, 2008.

[22] R. A. Schmidt and D. Tishkovsky. Automated synthesis of tableau calculi. In *TABLEAUX09*, vol. 5607 of *Lect. Notes Artif. Intell.*, pp. 310–324. Springer, 2009.

[23] F. Wolter and M. Zakharyaschev. Undecidability of the unification and admissibility problems for modal and description logics. *ACM Transdactions on Computational Logic*, 9(4):1–20, 2008.

[24] D. Zucchelli. Studio e realizzazione di algoritmi per l'unificazione nelle logiche modali. Laurea specialistica in informatica (Masters Thesis), Università degli Studi di Milano, 2004. In Italian. Supervisor: Silvio Ghilardi.