Two Tableau-Based Decision Procedures for Hybrid Logic

Thomas Bolander

Dep. Informatics and Mathematical Modelling
Building 322
Technical University of Denmark
DK-2800 Lyngby, Denmark
`tb@imm.dtu.dk`

Torben Braüner

Department of Computer Science
Roskilde University
P.O. Box 260
DK-4000 Roskilde, Denmark
`torben@ruc.dk`

Abstract

It is well-known that various hybrid logics without binders are decidable, but decision procedures are usually not based on tableau systems. In this paper we give two tableau-based decision procedures for a very expressive hybrid logic including the universal modality. The decision procedures make use of so-called loop-checks which is a technique standardly used in connection with tableau systems for other logics, namely prefixed tableau systems for transitive modal logics, as well as prefixed tableau systems for certain description logics.

Keywords: Hybrid logic, modal logic, universal modality, tableau systems, decision procedures.

1 Introduction

The hybrid logic we consider in the present paper is obtained by adding to ordinary modal logic further expressive power in the form of a second sort of propositional symbols called nominals, and moreover, by adding so-called satisfaction operators as well as the universal modality. A nominal is assumed to be true at exactly one world, so in this sense a nominal refers to a world. If a is a nominal and ϕ is an arbitrary formula, then a new formula $a : \phi$ called a satisfaction statement can be formed. The part a: of $a : \phi$ is called a satisfaction operator (some authors often use the notation $@_a$ instead of a:). The satisfaction statement $a:\phi$ is true (at any world) if and only if the formula ϕ is true at one particular world, namely the world at which the nominal a is true. The truth-condition of the universal modality E is that $E\phi$ is true (at any world) if and only if there exists a world at which the formula ϕ is true.

It is well-known that the hybrid logic described above is decidable, see [1], but decision procedures are usually not tableau-based. In fact, we are only aware of one published tableau-based decision procedure for hybrid logic, namely the one given in Miroslava Tzakova's paper [11]. However, a number of crucial details are missing in Tzakova's termination proof, and we did not find any way to fill out these details. In the present paper we give a tableau system along the lines of Tzakova's system extended with the universal modality, and give a terminating systematic tableau construction algorithm for the system. Our tableau construction algorithm is very different from Tzakova's algorithm. An essential feature of our algorithm is that it makes use of loop-checks. Besides Tzakova's tableau system, we also consider a tableau system given by Patrick Blackburn in the paper [2]. Decision procedures are not considered in Blackburn's paper. We give a terminating systematic tableau construction algorithm for Blackburn's system extended with the universal modality, again with the essential feature that it makes use of loop-checks. Analogous results follows for the weaker hybrid logic obtained by ignoring the universal modality.

The paper is structured as follows. In the second section we recapitulate the basics of hybrid logic, in the third section we give the decision procedure for our version of Tzakova's tableau system, and in the fourth section we give the decision procedure for Blackburn's tableau system. In the final section we discuss some related work.

2 The basics of hybrid logic

We shall in many cases adopt the terminology of [3] and [1]. The hybrid logic we consider is obtained by adding a second sort of propositional symbols called *nominals* to ordinary modal logic. It is assumed that a set of ordinary propositional symbols and a countably infinite set of nominals are given. The sets are assumed to be disjoint. The metavariables $p, q, r, \ldots$ range over ordinary propositional symbols and $a, b, c, \ldots$ range over nominals. Besides nominals, an operator a: called a *satisfaction operator* is added for each nominal a, and furthermore, the universal modality E is added. The formulas of hybrid modal logic are defined by the grammar

$$S \ ::= \ p \mid a \mid \neg S \mid S \wedge S \mid \Diamond S \mid a : S \mid ES$$

where p is an ordinary propositional symbol and a is a nominal. In what follows, the metavariables ϕ, ψ, χ, ...range over formulas. Formulas of the form $a : \phi$ are called *satisfaction statements*, cf. a similar notion in [2]. The operator $\Box$ and the propositional connectives not taken as primitive are defined as usual.

We now define models.

Definition 2.1 A *model* for hybrid logic is a tuple (W, R, V) where

(i) W is a non-empty set;

(ii) R is a binary relation on W; and

(iii) V is a function that to each pair consisting of an element of W and an ordinary propositional symbol assigns an element of $\{0, 1\}$.

The elements of W are called *worlds* and the relation R is called an *accessibility relation*. An *assignment* for a model $\mathcal{M} = (W, R, V)$ is a function g that to each nominal assigns an element of W. Given assignments g' and g, $g' \overset{a}{\sim} g$ means that g' agrees with g on all nominals save possibly a. The relation $\mathcal{M}, g, w \models \phi$ is defined inductively, where g is an assignment, w is an element of W, and ϕ is a formula.

$$\mathcal{M}, g, w \models p \text{ iff } V(w, p) = 1$$

$$\mathcal{M}, g, w \models a \text{ iff } w = g(a)$$

$$\mathcal{M}, g, w \models \neg\phi \text{ iff not } \mathcal{M}, g, w \models \phi$$

$$\mathcal{M}, g, w \models \phi \wedge \psi \text{ iff } \mathcal{M}, g, w \models \phi \text{ and } \mathcal{M}, g, w \models \psi$$

$$\mathcal{M}, g, w \models a : \phi \text{ iff } \mathcal{M}, g, g(a) \models \phi$$

$$\mathcal{M}, g, w \models \Diamond\phi \text{ iff for some } v \in W,\ wRv \text{ and } \mathcal{M}, g, v \models \phi$$

$$\mathcal{M}, g, w \models E\phi \text{ iff for some } v \in W,\ \mathcal{M}, g, v \models \phi$$

By convention $\mathcal{M}, g \models \phi$ means $\mathcal{M}, g, w \models \phi$ for every element w of W and $\mathcal{M} \models \phi$ means $\mathcal{M}, g \models \phi$ for every assignment g. A formula ϕ is *valid* if and only if $\mathcal{M} \models \phi$ for any model $\mathcal{M}$.

3 Tzakova's system extended with the universal modality

Tzakova's system [11] is a prefixed tableau calculus (see the book [4] for the basics of tableau systems). This means that the formulas occurring in the tableau rules are *prefixed formulas* on the form $\sigma\phi$, where ϕ is a formula of hybrid modal logic and σ belongs to some fixed countably infinite set of symbols called *prefixes*. In addition, the tableau rules contain *accessibility formulas* on the form $\sigma < \sigma'$ where σ and σ' are prefixes. The rules of the tableau system are given in Figure 1. Actually, the given tableau system is a modified version of Tzakova's calculus. The calculus is simplified by replacing Tzakova's rules (S-Identifying) and (L-Identifying) by (Id). Furthermore, the rule (Labeling) has been deleted. Our calculus also differs from Tzakova's by including the rules for the universal modality. However, we will still refer to this calculus as *Tzakova's system*. A *tableau* in Tzakova's system is a well-

$$\frac{\sigma\neg\neg\phi}{\sigma\phi}\,(\neg\neg)$$

$$\frac{\sigma(\phi\wedge\psi)}{\sigma\phi,\sigma\psi}\,(\wedge) \qquad \frac{\sigma\neg(\phi\wedge\psi)}{\sigma\neg\phi \mid \sigma\neg\psi}\,(\neg\wedge)$$

$$\frac{\sigma c:\phi}{\sigma' c,\sigma'\phi}\,(:)^{*} \qquad \frac{\sigma\neg c:\phi}{\sigma' c,\sigma'\neg\phi}\,(\neg:)^{*}$$

$$\frac{\sigma\Diamond\phi}{\sigma'\phi,\sigma<\sigma'}\,(\Diamond)^{*} \qquad \frac{\sigma\neg\Diamond\phi,\sigma<\sigma'}{\sigma'\neg\phi}\,(\neg\Diamond)$$

$$\frac{\sigma E\phi}{\sigma'\phi}\,(E)^{*} \qquad \frac{\sigma\neg E\phi}{\sigma''\neg\phi}\,(\neg E)^{\dagger}$$

$$\frac{\sigma\phi,\sigma c,\tau c}{\tau\phi}\,(Id)$$

$*$ The prefix σ' is new to the tableau.
$\dagger$ The prefix σ'' is on the branch.

Fig. 1. Modified version of Tzakova's tableau rules

founded tree in which each node is either a prefixed formula or an accessibility formula, and the edges represent applications of tableau rules in the usual way. We impose the following conventions on the application of the rules in tableau constructions.

- In constructing a tableau, the rules (:), ($\neg$:), ($\Diamond$) and (E) are never applied to the same premise twice on the same branch.
- A formula is never added to a tableau branch where it already occurs.

3.1 Some properties of the system

Tzakova's system satisfies the following basic properties.

Lemma 3.1 (Quasi-subformula property) *If a formula $\sigma\phi$ occurs in a tableau with root $\sigma_0\phi_0$ then either ϕ or $\neg\phi$ is a subformula of ϕ_0.*

Proof. Follows immediately from the rules in Figure 1. □

Note the following consequence of Lemma 3.1: For any given tableau $\mathcal{T}$, the set $\{\phi \mid \sigma\phi \text{ occurs in } \mathcal{T}\}$ is finite. We will use this fact a number of times in the proofs below.

The only way new prefixes can be introduced to a tableau is by using one of the rules (:), (¬:), (◊) or (E). These introduce a new prefix σ' from a given prefix σ. Let Θ be a branch of a tableau. If a new prefix σ' is introduced by applying one of the rules (:), (¬:), (◊) or (E) to a prefixed formula $\sigma\phi$ then we say that σ' is *generated* by σ with respect to Θ, and we write $\sigma <_\Theta \sigma'$. This gives us a binary relation $<_\Theta$ on the prefixes occurring on Θ.

Proposition 3.2 *Let Θ be a branch of a tableau. Let N^Θ be the set of prefixes occurring on Θ. The graph $(N^\Theta, <_\Theta)$ is a finitely branching tree.*

Proof. That the graph is a tree follows from the fact that each prefix in N^Θ can be generated by at most one other prefix, and that all prefixes in N^Θ must have the prefix of the root formula as an ancestor. That the graph is finitely branching follows from the fact that for any given prefix σ the set $\{\phi \mid \sigma\phi \text{ occurs on } \Theta\}$ is finite (cf. Lemma 3.1), and each of these finitely many formulas $\sigma\phi$ can generate at most one new successor prefix σ' (by applying one of the rules (:), (¬:), (◊) or (E) to $\sigma\phi$). □

3.2 Systematic tableau construction

Before giving the systematic tableau construction algorithm we need a definition.

Definition 3.3 Let σ and τ be prefixes occurring at a branch Θ of a tableau. The prefix σ is *included* in the prefix τ with respect to Θ if for any hybrid formula ϕ, if $\sigma\phi$ occurs on Θ then $\tau\phi$ also occurs on Θ. If σ is included in τ and τ has its first occurrence on Θ no later than σ, then we write $\sigma \subseteq_\Theta \tau$.

We are now ready to define the systematic tableau construction algorithm. The algorithm we present is non-deterministic, but can easily be made deterministic by introducing suitable well-orderings.

Definition 3.4 (Tableau construction algorithm) Let ϕ be the formula whose validity we have to decide. By induction we define a sequence $\mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \ldots$ of finite tableaus, where each tableau is obtained from the previous by applying one of the tableau rules. Define $\mathcal{T}_0$ to be the tableau constituted by the single prefixed formula $\sigma\neg\phi$, where σ is any prefix. Given a tableau $\mathcal{T}_i$, we then define $\mathcal{T}_{i+1}$ to be the tableau obtained by applying some rule to $\mathcal{T}_i$ subject to the following restriction:

($\mathcal{R}$) The rules (:), (¬:), (◊) and (E) are not applied to a formula occurrence $\sigma\phi$ at a branch Θ of $\mathcal{T}_i$ if there exists a prefix τ such that $\sigma \subseteq_\Theta \tau$.

If no rule applies satisfying restriction $\mathcal{R}$, the algorithm is terminated.

Theorem 3.5 *The systematic tableau construction algorithm terminates.*

Proof. Assume to obtain a contradiction that this is not the case. Then the tableau $\cup_{i\in\omega}\mathcal{T}_i$ must be infinite. Thus it contains an infinite path Θ. By the tableau conventions, all prefixed formulas along this path are distinct. Using Lemma 3.1, it follows that Θ must contain infinitely many different prefixes. Therefore the graph $(N^{\Theta}, <_{\Theta})$ must be infinite. Since by Proposition 3.2 the graph is a finitely branching tree, it must contain an infinite path $\sigma_1 <_{\Theta} \sigma_2 <_{\Theta} \sigma_3 <_{\Theta} \cdots$. For each $i > 0$, let Θ_i be the initial segment of Θ up to, but not including, the formula containing the first occurrence of σ_{i+1}. Let Γ_i be the set $\Gamma_i = \{\phi \mid \sigma_i\phi \text{ occurs at } \Theta_i\}$. All Γ_i contain only formulas that are either subformulas of the root formula or negations of such formulas (Lemma 3.1). Since there are only finitely many such formulas, not all Γ_i can be distinct. In other words, there exists i, j with $i < j$ such that $\Gamma_i = \Gamma_j$. We will now prove that $\sigma_j \subseteq_{\Theta_j} \sigma_i$. Since i is less than j, the first occurrence of σ_i on Θ_j must come before the first occurrence of σ_j. Let ϕ be an arbitrary formula for which $\sigma_j\phi$ occurs on Θ_j, that is, $\phi \in \Gamma_j$. Since $\Gamma_i = \Gamma_j$, we have that $\sigma_i\phi$ occurs on Θ_i, and since Θ_i is an initial segment of Θ_j, we get that $\sigma_i\phi$ occurs on Θ_j. This proves that $\sigma_j \subseteq_{\Theta_j} \sigma_i$. Now consider the first formula containing an occurrence of σ_{j+1}. By definition, this is the first formula not on Θ_j, so it must be introduced by applying some rule to a formula occurrence at Θ_j. The prefix σ_{j+1} is generated by σ_j, so σ_{j+1} is introduced by applying one of the rules (:), ($\neg$:), ($\Diamond$) or (E) to a formula $\sigma_j\psi$ at Θ_j. However, this is in contradiction with restriction $\mathcal{R}$ by which none of the rules (:), ($\neg$:), ($\Diamond$) or (E) can be applied to the formula $\sigma_j\psi$ at Θ_j since $\sigma_j \subseteq_{\Theta_j} \sigma_i$. □

3.3 Soundness and completeness

Soundness of the tableau calculus in Figure 1 can be proved by showing that each rule preserves satisfiability [11]. The only rules in our calculus which are not already covered by Tzakova's system are (Id), (E) and ($\neg E$). It is simple to prove that these rules preserve satisfiability in hybrid models. We now turn to the completeness proof. To prove completeness of the systematic tableau construction algorithm it is sufficient to prove that if a tableau with root $\sigma_0\phi_0$ has an open branch Θ then there exists a model $\mathcal{M}_{\Theta}$, an assignment g and a world w such that $\mathcal{M}_{\Theta}, g, w \models \phi_0$ holds. We will now describe how $\mathcal{M}_{\Theta}$ is constructed from an open tableau branch Θ. First a little extra machinery.

Definition 3.6 (Urfathers) Let Θ be a branch of a tableau and let σ be a prefix occurring on Θ. We define the *urfather* of σ with respect to Θ to be the earliest occurring prefix on Θ which σ is included in.[1] The urfather of σ with respect to Θ is denoted $u_{\Theta}(\sigma)$. Prefixes σ on Θ for which $u_{\Theta}(\sigma) = \sigma$ are called *urfathers* on Θ.

Lemma 3.7 *Let $\mathcal{T}$ be a tableau obtained from the tableau construction algorithm. $\mathcal{T}$ is closed under each of the rules ($\neg\neg$), ($\wedge$), ($\neg\wedge$), ($\neg\Diamond$), ($\neg E$) and (Id) of Figure 1. Furthermore, $\mathcal{T}$ is closed under the rules (:), ($\neg$:), ($\Diamond$) and*

[1] That is, the *urfather* of σ wrt. Θ is the unique prefix σ' satisfying: (i) $\sigma \subseteq_{\Theta} \sigma'$; (ii) there is no σ'' s.t. $\sigma' \subseteq_{\Theta} \sigma''$.

(E) *whenever the premise is a formula occurrence of the form* $\sigma\phi$ *where* σ *is an urfather on the branch containing the occurrence.*

Proof. Consider the sequence of tableaus constructed by the tableau algorithm leading to $\mathcal{T}$. Since the algorithm terminates, this must be a finite sequence $\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_n$ where $\mathcal{T} = \mathcal{T}_n$. By definition, no rule applies to $\mathcal{T}_n$ that satisfies restriction $\mathcal{R}$. Since $\mathcal{R}$ only concerns the rules $(:)$, $(\neg:)$, $(\Diamond)$ and (E) we immediately get that the tableau is closed under all rules except possibly these. Now consider the rule $(\Diamond)$. Assume a branch Θ of $\mathcal{T}_n$ contains $\sigma\Diamond\phi$ where σ is an urfather on Θ. By definition of $\mathcal{T}_n$, no rule applies to $\sigma\Diamond\phi$ that satisfies $\mathcal{R}$. However, since σ is an urfather on Θ there is no prefix τ such that $\sigma \subseteq_\Theta \tau$. Thus the rule $(\Diamond)$ is not blocked by restriction $\mathcal{R}$. The only possible reason that the rule $(\Diamond)$ can not be applied to $\sigma\Diamond\phi$ on $\mathcal{T}_n$ is therefore that it has already been applied earlier in the tableau construction (cf. the tableau convention introduced in the beginning of Section 3). This proves closure under the rule $(\Diamond)$. Closure under the rules $(:)$, $(\neg:)$ and (E) are proved similarly. □

Lemma 3.8 *Let* Θ *be a branch of a tableau and let* σ *and* τ *be prefixes occurring on* Θ*. If there exists a nominal* c *such that both* σc *and* τc *occurs on* Θ *then* σ *is included in* τ *with respect to* Θ*.*

Proof. Let $\sigma\phi$ be a formula occurring on Θ. We have to prove that $\tau\phi$ occurs on Θ as well. This follows immediately from Lemma 3.7, since Θ contains all of $\sigma\phi$, σc and τc and is closed under the rule (Id). □

Given a tableau branch Θ, we define the model $\mathcal{M}_\Theta$ by

$$\begin{aligned}
\mathcal{M}_\Theta &= (W_\Theta, R_\Theta, V_\Theta), \text{ where} \\
W_\Theta &= \{u_\Theta(\sigma) \mid \sigma \text{ occurs on } \Theta\} \\
R_\Theta &= \{(\sigma, u_\Theta(\tau)) \in W_\Theta^2 \mid \sigma < \tau \text{ occurs on } \Theta\} \\
V_\Theta(\sigma, p) &= 1 \text{ iff } \sigma p \text{ occurs on } \Theta.
\end{aligned}$$

Furthermore, we define an assignment g_Θ for $\mathcal{M}_\Theta$ in the following way. We let $g_\Theta(c)$ be the prefix of the root formula of Θ if there is no prefix σ on Θ such that σc occurs on Θ (note that the prefix of the root formula is always an urfather). Otherwise we let $g_\Theta(c)$ be the urfather of the prefixes σ for which σc occurs on Θ. This defines $g_\Theta(c)$ uniquely, since if σ and σ' are prefixes such that both σc and $\sigma' c$ occurs on Θ, then it follows from Lemma 3.8 that σ will be included in σ' and they must thus have the same urfather. We are now ready to prove the completeness theorem. As mentioned above, it suffices to prove that if a tableau with root $\sigma_0\phi_0$ has an open branch Θ then there is a world w such that $\mathcal{M}_\Theta, g_\Theta, w \models \phi_0$. What we will prove is slightly stronger.

Theorem 3.9 (Completeness) *Let* Θ *be an open branch of a tableau constructed using the tableau algorithm of Section 3.2. For any prefixed formula* $\sigma\phi$ *on* Θ *where* σ *is an urfather on* Θ *we have* $\mathcal{M}_\Theta, g_\Theta, \sigma \models \phi$*.*

Proof. The proof is by induction on the structure of ϕ. First assume σp occurs on Θ where p is a propositional symbol and σ is an urfather. Then $V_\Theta(\sigma, p) = 1$ and thus $\mathcal{M}_\Theta, g_\Theta, \sigma \models p$ as needed. Now assume σc occurs on Θ where c is a nominal and σ is an urfather. Then $g_\Theta(c) = \sigma$, by definition of g_Θ, and thus $\mathcal{M}_\Theta, g_\Theta, \sigma \models c$, as needed. This covers the base case. We now turn to the induction step.

Consider the case where $\sigma\neg\neg\psi$ occurs on Θ and σ is an urfather. By closure under the rule $(\neg\neg)$ (Lemma 3.7) it follows that $\sigma\psi$ occurs on Θ as well. From the induction hypothesis we get $\mathcal{M}_\Theta, g_\Theta, \sigma \models \psi$, and thus $\mathcal{M}_\Theta, g_\Theta, \sigma \models \neg\neg\psi$ immediately follows. The other propositional cases $\sigma\psi \wedge \chi$ and $\sigma\neg(\psi \wedge \chi)$ are treated similarly.

Consider the case where $\sigma c : \psi$ occurs on Θ and σ is an urfather. By closure under the rule $(:)$ (Lemma 3.7), there exists a prefix σ' such that $\sigma' c$ and $\sigma'\psi$ also occurs on Θ. Let $\sigma'' = g_\Theta(c)$. Then σ'' is the urfather of σ' on Θ. From this it follows that $\sigma''\psi$ occurs on Θ as well. By induction hypothesis it follows that $\mathcal{M}_\Theta, g_\Theta, \sigma'' \models \psi$. Since $\sigma'' = g_\Theta(c)$ this proves $\mathcal{M}_\Theta, g_\Theta, \sigma \models c : \psi$, as needed. The case $\sigma\neg c : \psi$ is proved similarly.

Consider the case where $\sigma\Diamond\psi$ occurs on Θ and σ is an urfather. By closure under the rule $(\Diamond)$ (Lemma 3.7), there exists a prefix σ' such that both $\sigma'\psi$ and $\sigma < \sigma'$ occurs on Θ. Let $\sigma'' = u_\Theta(\sigma')$. The induction hypothesis gives $\mathcal{M}_\Theta, g_\Theta, \sigma'' \models \psi$. Since $\sigma < \sigma'$ occurs on Θ we have that R_Θ contains the pair $(\sigma, u_\Theta(\sigma')) = (\sigma, \sigma'')$. Thus we get $\mathcal{M}_\Theta, g_\Theta, \sigma \models \Diamond\psi$.

Consider the case where $\sigma\neg\Diamond\psi$ occurs on Θ and σ is an urfather. We have to prove $\mathcal{M}_\Theta, g_\Theta, \sigma \not\models \Diamond\psi$. If there is no prefix τ such that $\sigma R_\Theta \tau$ then this trivially holds. Otherwise, let τ be any prefix with $\sigma R_\Theta \tau$. We have to prove $\mathcal{M}_\Theta, g_\Theta, \tau \models \neg\psi$. By definition of R_Θ, τ is the urfather of a prefix τ' such that $\sigma < \tau'$ occurs on Θ. Since both $\sigma\neg\Diamond\psi$ and $\sigma < \tau'$ occurs on Θ, we get by closure under the rule $(\neg\Diamond)$ (Lemma 3.7) that $\tau'\neg\psi$ occurs on Θ as well. Since τ is the urfather of τ', the formula $\tau\neg\psi$ must also occur on Θ. By induction hypothesis we then have $\mathcal{M}_\Theta, g_\Theta, \tau \models \neg\psi$, as needed.

Consider the case where $\sigma E\psi$ occurs on Θ and σ is an urfather. By closure under the rule (E) (Lemma 3.7) there exists a prefix σ' such that $\sigma'\psi$ occurs on Θ. Let σ'' be the urfather of σ' on Θ. Then $\sigma''\psi$ also occurs on Θ and by induction hypothesis we get $\mathcal{M}_\Theta, g_\Theta, \sigma'' \models \psi$. This proves $\mathcal{M}_\Theta, g_\Theta, \sigma \models E\psi$.

Finally consider the case where $\sigma\neg E\psi$ occurs on Θ and σ is an urfather. We have to prove $\mathcal{M}_\Theta, g_\Theta, \sigma \models \neg E\psi$, that is, for all $\sigma' \in W_\Theta$, $\mathcal{M}_\Theta, g_\Theta, \sigma' \models \neg\psi$. To prove this, let an arbitrary element σ' in W_Θ be chosen. The element σ' is an urfather occurring on the branch Θ. By closure under the rule $(\neg E)$ (Lemma 3.7), $\sigma'\neg\psi$ occurs on Θ. Thus the induction hypothesis gives us $\mathcal{M}_\Theta, g_\Theta, \sigma' \models \neg\psi$ as needed.

□

$$\frac{a : \neg\phi}{\neg a : \phi}\,(\neg) \qquad \frac{\neg a : \neg\phi}{a : \phi}\,(\neg\neg)$$

$$\frac{a : (\phi \wedge \psi)}{a : \phi, a : \psi}\,(\wedge) \qquad \frac{\neg a : (\phi \wedge \psi)}{\neg a : \phi \mid \neg a : \psi}\,(\neg\wedge)$$

$$\frac{a : b : \phi}{b : \phi}\,(:) \qquad \frac{\neg a : b : \phi}{\neg b : \phi}\,(\neg:)$$

$$\frac{a : \Diamond\phi}{c : \phi, a : \Diamond c}\,(\Diamond)^{*\star} \qquad \frac{\neg a : \Diamond\phi, a : \Diamond d}{\neg d : \phi}\,(\neg\Diamond)$$

$$\frac{a : E\phi}{c : \phi}\,(E)^{*} \qquad \frac{\neg a : E\phi}{\neg d : \phi}\,(\neg E)^{\dagger}$$

$$\frac{}{d : d}\,(Ref)^{\dagger} \qquad \frac{a : b}{b : a}\,(Sym)$$

$$\frac{a : b, b : \phi}{a : \phi}\,(Nom1)^{\ddagger} \qquad \frac{a : b, b : \Diamond c}{a : \Diamond c}\,(Nom2) \qquad \frac{a : \Diamond b, b : c}{a : \Diamond c}\,(Bridge)$$

$*$ The nominal c is new.
$\star$ The formula ϕ is not a nominal.
$\dagger$ The nominal d is on the branch.
$\ddagger$ ϕ is a propositional symbol (ordinary or a nominal).

Fig. 2. Blackburn's tableau rules and rules for the universal modality

4 Blackburn's system extended with the universal modality

The tableau system considered in the present section is a slightly modified, and also extended, version of a system originally given in the paper [2] by Patrick Blackburn. The rules are given in Figure 2. The rules are identical to the rules given in [2] except that in his system the rules for the universal modality are not included, and moreover, in his system the rule ($Nom1$) is not restricted to propositional symbols, and consequently, the rule ($Nom2$) is omitted. It turns out that we do not need the more general version of ($Nom1$) given in [2] and restricting it as we have done here simplifies later technical considerations. We have taken the connectives $\vee$ and $\Box$ to be defined, not primitive, so they do not need separate rules. All formulas in the rules are satisfaction statements. A *tableau* in the system is a well-founded tree in which each node is a satisfaction statement and the edges represent applications of tableau rules in the usual way. When it is appropriate, we shall often blur the distinction between a formula and an occurrence of the formula in a tableau.

We shall make use of some important conventions about the rules of Figure 2. The rules $(\neg)$, $(\neg\neg)$, $(\wedge)$, $(\neg\wedge)$, $(:)$, $(\neg:)$, $(\Diamond)$, and (E) will be called *destructive* rules and the remaining rules will be called *non-destructive.* Note that a destructive rule has exactly one formula in the premise. The destructive rules $(\Diamond)$ and (E) will also be called *existential.* Thus, we have three categories of rules: Destructive rules which are not existential, existential destructive rules, and non-destructive rules. The rules of these three categories are applied in different ways.

- A destructive rule which is not existential is applied to a formula occurrence ϕ in a tableau by extending all branches through ϕ in accordance with the rule. After the application, it is recorded that the rule was applied to ϕ and the rule will not again be applied to ϕ.
- An existential destructive rule is applied to a formula occurrence ϕ on a branch Θ by extending Θ in accordance with the rule. After the application, it is recorded that the rule was applied to ϕ with respect to Θ and the rule will not again be applied to ϕ with respect to Θ or any extension of Θ.
- A non-destructive rule is applied to a set of formula occurrences (note that a non-destructive rule has zero, one, or two formulas in the premise) on a branch Θ by extending Θ in accordance with the rule. No information is recorded about applications of non-destructive rules.

Both destructive and non-destructive rules are subject to the exception that if a formula to be added to a branch is already present on the branch, then the addition of the formula is simply omitted. It follows that a formula cannot occur more than once at a branch.

Note that non-destructive rules are only applicable to formulas of the forms $a : p$, $a : c$, $a : \Diamond c$, $\neg a : \Diamond\phi$, and $\neg a : E\phi$ and conversely, destructive rules are only applicable to formulas not of these forms (in fact, exactly one destructive rule is applicable to any formula which is not of one of these forms). So the classification of rules as destructive and non-destructive corresponds to a classification of formulas. Also note that when applying a destructive rule which is not existential, it is applied to a formula occurrence with respect to all branches through it, but when applying an existential rule, the rule is applied with respect to one particular branch. This more fine-grained applicability of existential rules is a prerequisite for being able to incorporate so-called loop-checks, cf. Definition 4.11.

4.1 Some properties of the system

The tableau system satisfies the following important property, which is similar to the well-known subformula property of the standard propositional tableau system.

Lemma 4.1 *(Quasi-subformula property) If a formula $a{:}\phi$ occurs in a tableau where ϕ is not a nominal and ϕ is not of the form $\Diamond b$, then ϕ is a positively occurring subformula of the root formula. If a formula $\neg a : \phi$ occurs in a tableau, then ϕ is a negatively occurring subformula of the root formula.*

Proof. A simultaneous induction where each rule is checked. □

Below we shall give some further results which shows some interesting features of the tableau system. First two definitions.

Definition 4.2 Let Θ be a branch of a tableau and let N^Θ be the set of nominals occurring in the formulas of Θ. Define a binary relation $\sim_\Theta$ on N^Θ by $a \sim_\Theta b$ if and only if the formula $a{:}b$ occurs at Θ. Let $\sim^*_\Theta$ be the reflexive, symmetric, and transitive closure of $\sim_\Theta$.

Definition 4.3 An occurrence of a nominal in a formula is *equational* if the occurrence is a formula (that is, if it is not part of a satisfaction operator).

For example, the occurrence of the nominal c in the formula $\phi \wedge c$ is equational but the occurrence of c in $\psi \wedge c : \chi$ is not. The justification for this terminology is that a nominal in the first-order correspondence language (and thereby also in the semantics) gives rise to an equality statement if and only if the nominal occurrence in question occurs equationally. The theorem below will be used later in the completeness theorem, Theorem 4.17.

Theorem 4.4 *Let $a : b$ be a formula occurrence on a branch Θ of a tableau. If the nominals a and b are different, then each of them has the property that it is identical to, or related by $\sim_\Theta$ to, a nominal with a positive and equational occurrence in the root formula.*

Proof. Check each rule. Lemma 4.1 is needed in a number of the cases. In the case with the rule $(\Diamond)$, we make use of the restriction that the rule cannot be applied to formulas of the form $a : \Diamond\phi$ where ϕ is a nominal. □

Corollary 4.5 *Let Θ be a branch of a tableau. Any non-singleton equivalence class wrt. the equivalence relation $\sim^*_\Theta$ contains a nominal which occurs positive and equational in the root formula.*

Proof. Follows directly from Theorem 4.4. □

We think the corollary above is of independent interest. It says that non-trivial equational reasoning, that is, reasoning involving non-singleton equivalence classes, only takes place in connection with certain nominals in the root formula, namely those that occur positive and equational. Note that this implies that pure modal input to the tableau only gives rise to reasoning involving singleton equivalence classes.

Definition 4.6 A formula occurrence on a branch of a tableau is an *accessibility* formula occurrence if it is an occurrence of the formula $a : \Diamond c$ generated by the rule $(\Diamond)$.

Note that if the rule $(\Diamond)$ is applied to a formula occurrence $a{:}\Diamond\Diamond b$, resulting in the branch being extended with $a : \Diamond c$ and $c : \Diamond b$, then the occurrence of $a : \Diamond c$ is an accessibility formula occurrence, but the occurrence of $c : \Diamond b$ is not. The theorem below will be used later in the completeness theorem, Theorem 4.17.

Theorem 4.7 *Let $a : \Diamond b$ be a formula occurrence on a branch Θ of a tableau. Either there is a positively occurring subformula $\Diamond b'$ of the root formula such that $b \sim^*_\Theta b'$ or there is an accessibility formula occurrence $a' : \Diamond b'$ at Θ such that $a \sim^*_\Theta a'$ and $b \sim^*_\Theta b'$.*

Proof. Check each rule. Lemma 4.1 is needed in some of the cases. □

The only way new nominals can be introduced to a tableau is by using one of the rules $(\Diamond)$ or (E) which we called existential rules. This motivates the following definition.

Definition 4.8 Let Θ be a branch of a tableau. If a new nominal c is generated by applying an existential rule to a satisfaction statement $a : \phi$, then we write $a <_\Theta c$.

The definition above gives us a binary relation $<_\Theta$ on the set N^Θ.

Proposition 4.9 *Let Θ be a branch of a tableau. The graph $(N^\Theta, <_\Theta)$ is the disjoint union of a finite set of finitely branching trees.*

Proof. That the relation is the disjoint union of a set of trees follows from the observation that if $a <_\Theta c$, then the nominal c is new. That the set of trees is finite follows the observation that for any new nominal c there is a nominal a such that $a <_\Theta c$, thus, the nominal c cannot be the root of a tree. The following argument shows that the trees are finitely branching. Assume conversely that there exists an infinite sequence $a <_\Theta c_1$, $a <_\Theta c_2, \ldots$ of edges. For each i, the edge $a <_\Theta c_i$ is generated by applying an existential rule to some formula occurrence χ_i. Consider the sequence χ_1, $\chi_2, \ldots$ of formula occurrences. The existential rules are destructive, so the formula occurrences in this sequence are distinct, and moreover, a formula cannot occur more than once at a branch. It follows that the formula occurrences in the sequence χ_1, $\chi_2, \ldots$ are occurrences of infinitely many different formulas. Now, if the edge $a <_\Theta c_i$ is generated by applying the existential rule $(\Diamond)$ to χ_i, then χ_i is of the form $a : \Diamond\phi_i$ where ϕ_i is not a nominal, and hence, $\Diamond\phi_i$ is a subformula of the root formula by Lemma 4.1, and if $a <_\Theta c_i$ is generated by applying the other existential rule (E) to χ_i, then χ_i is of the form $a : E\phi_i$, and hence, $E\phi_i$ is a subformula of the root formula, again by Lemma 4.1. But there are only finitely many subformulas of the root formula, which contradicts that infinitely many different formulas occur in the the sequence χ_1, χ_2, ... □

Note that in the above results we have not made any assumptions on which rules are applied at the branch Θ, but if we assume that Θ is closed under the rules (*Ref*), (*Sym*), and (*Nom*1), then $\sim_\Theta^*$ coincides with $\sim_\Theta$.

4.2 Systematic tableau construction

Before giving the systematic tableau construction algorithm, we need a definition.

Definition 4.10 Let b and a be nominals occurring at a branch Θ of a tableau. The nominal a is *included* in the nominal b with respect to Θ if for any subformula ϕ of the root formula, if the formula $a : \phi$ occurs on Θ, then $b : \phi$ also occurs on Θ, and similarly, if $\neg a : \phi$ occurs on Θ, then $\neg b : \phi$ also occurs on Θ. If a is included in b with respect to Θ, and the first occurrence of b on Θ is before the first occurrence of a, then we write $a \subseteq_\Theta b$.

We are now ready to give the systematic tableau construction algorithm.

Definition 4.11 Let $a{:}\phi$ be the formula whose validity we have to decide. We define by induction a sequence $\mathcal{T}_0$, $\mathcal{T}_1$, $\mathcal{T}_2$, ... of finite tableaus, each of which is embedded in all its successors. Let $\mathcal{T}_0$ be the finite tableau constituted by the single unmarked formula $\neg a : \phi$. Assume that the finite tableau $\mathcal{T}_n$ is defined. If possible, apply an arbitrary rule with the following restrictions:

- (i) The rule ($\Diamond$) is not applied to a formula occurrence $a : \Diamond\phi$ at a branch Θ if there exists a nominal b such that $a \subseteq_\Theta b$.
- (ii) The rule (E) is not applied to a formula occurrence $a : E\phi$ at a branch Θ if there exists a nominal b such that $a \subseteq_\Theta b$.

Let $\mathcal{T}_{n+1}$ be the resulting tableau.

The conditions on applications of rules are so-called loop-check conditions. The intuition behind loop-checks is that an existential rule is not applied in a world if the information in that world can be found already in an ancestor world. Hence, the generation of a new world by the existential rule is blocked.

We shall now prove that the algorithm always terminates in the sense that there always exists an n such that $\mathcal{T}_n = \mathcal{T}_{n+1}$.

Theorem 4.12 *The systematic tableau construction algorithm terminates.*

Proof. Assume conversely that the algorithm does not terminate. Then the resulting tableau is infinite, and hence, has an infinite branch Θ. The graph $(N^\Theta, <_\Theta)$ is the disjoint union of a finite set of finitely branching trees cf. Proposition 4.9, so it has an infinite branch $a_1 <_\Theta a_2 <_\Theta a_3, \ldots$ (otherwise $\mathcal{N}^\Theta$ would be finite, and hence, by Lemma 4.1 there would only be finitely many formulas occurring at the branch Θ, contradicting that it is infinite). Now, for each i, let Θ_i be the initial segment of Θ up to, but not including, the first formula containing an occurrence of the nominal a_{i+1}. Thus, an existential rule was applied to a formula occurrence at the branch Θ_i resulting in the generation of a_{i+1}. Let Γ_i be the set of formulas which contains any subformula ϕ of the root formula such that $a_i : \phi$ occurs at the branch Θ_i, and similarly, let Δ_i be the set of formulas which contains any subformula ϕ of the root formula such that $\neg a_i : \phi$ occurs at the branch Θ_i. Since there are only finitely many sets of subformulas of the root formula, there exists j and k such that $j < k$ and $\Gamma_j = \Gamma_k$ as well as $\Delta_j = \Delta_k$. Clearly, the first occurrence of a_j on Θ is before the first occurrence of a_k. Moreover, for any subformula ϕ of the root formula, if $a_k : \phi$ occurs on Θ_k, then $\phi \in \Gamma_k$, and hence, $\phi \in \Gamma_j$, but then $a_j : \phi$ occurs on Θ_j which is an initial segment of Θ_k. A similar argument shows that if $\neg a_k : \phi$ occurs on Θ_k, then $\neg a_j : \phi$ also occurs on Θ_k. Hence, a_k is included in a_j with respect to Θ_k. We conclude that $a_k \subseteq_{\Theta_k} a_j$. But this contradicts that an existential rule was applied to a formula occurrence at the branch Θ_k resulting in the addition of the first formula containing an occurrence of the nominal a_{k+1}. Thus, the algorithm terminates. □

We have thus given a systematic tableau construction algorithm which gradually builds up a tableau and which terminates with a tableau having the property that no rules are applicable to it except for applications of existential rules blocked by the loop-check conditions.

4.3 Soundness and completeness

Soundness is straightforwardly obtained by extending the soundness proof of [2] with the universal modality. To prove completeness, we prove a model existence theorem. Throughout this subsection, we shall assume that Θ is a given branch of a tableau generated by the systematic tableau construction algorithm. Where no confusion can occur, we shall often omit reference to the branch Θ. First some machinery.

Definition 4.13 Let W be the subset of N^Θ containing any nominal a having the property that there is no nominal b such that $a \subseteq_\Theta b$. Let $\approx$ be the restriction of $\sim_\Theta$ to W.

Note that W contains all nominals of the root formula since the root formula is the first formula of the branch Θ. Observe that Θ is closed under the rules (*Ref*), (*Sym*), and (*Nom*1), so the relation $\sim_\Theta$ and hence also the relation $\approx$ are equivalence relations. Given a nominal a in W, we let $[a]_\approx$ denote the equivalence class of a with respect to $\approx$ and we let $W/\approx$ denote the set of equivalence classes.

Definition 4.14 Let R be the binary relation on W defined by aRc if and only if there exists nominals $a' \approx a$ and $c' \approx c$, satisfying one of the following three conditions.

(i) The formula $a' : \Diamond c'$ occurs at Θ is an accessibility formula occurrence.

(ii) There exists a nominal d in N^Θ such that the formula $a' : \Diamond d$ occurs at Θ as an accessibility formula occurrence and $d \subseteq_\Theta c'$.

(iii) The formula $a' : \Diamond c'$ occurs at Θ and a' or c' occurs in the root formula.

Note that the nominal d referred to in the second item in the definition is not an element of W. It is trivial that the relation R is compatible with $\approx$. We let $\overline{R}$ be the binary relation on $W/\approx$ defined by $[a]_\approx \overline{R}[c]_\approx$ if and only if aRc.

Definition 4.15 Let V be the function that to each pair consisting of an element of W and an ordinary propositional symbol assigns an element of $\{0, 1\}$ such that $V(a, p) = 1$ if $a : p$ occurs at Θ and $V(a, p) = 0$ otherwise.

It follows from Θ being closed under the rule $(Nom1)$ that V is compatible with $\approx$, so we let $\overline{V}$ be defined by $\overline{V}([a]_\approx, p) = V(a, p)$. We are now ready to define a model.

Definition 4.16 Let $\mathcal{M}$ be the model $(W/\approx, \overline{R}, \overline{V})$ and let the assignment g for $\mathcal{M}$ be defined by $g(a) = [a]_\approx$.

The model above is in some respects similar to the model defined in [2]. One crucial difference, however, is that the model above necessarily is finite.

Theorem 4.17 *Assume that the branch Θ is open, that is, if some formula $b:\chi$ occurs at Θ, then the formula $\neg b:\chi$ does not. For any formula $a:\phi$ which only contains nominals from W, the following two statements hold.*

- *If $a : \phi$ occurs at Θ, then it is the case that $\mathcal{M}, g, [a]_\approx \models \phi$.*
- *If $\neg a : \phi$ occurs at Θ, then it is not the case that $\mathcal{M}, g, [a]_\approx \models \phi$.*

Proof. Induction in the structure of ϕ. We only cover the most interesting case, namely where ϕ is of the form $\Diamond\psi$.

Assume that $a : \Diamond\psi$ occurs at Θ. We then have to prove that $\mathcal{M}, g, [a]_\approx \models \Diamond\psi$, that is, for some equivalence class $[c]_\approx$ such that $[a]_\approx \overline{R}[c]_\approx$, it is the case that $\mathcal{M}, g, [c]_\approx \models \psi$. We have two cases, according to whether the formula ψ is a nominal or not. We first consider the case where ψ is a nominal, say b. So we just have to prove that $[a]_\approx \overline{R}[b]_\approx$. By Theorem 4.7, either there is a nominal b' of the root formula such that $b' \sim_\Theta b$ or there is an accessibility formula occurrence $a' : \Diamond b'$ at Θ such that $a' \sim_\Theta a$ and $b' \sim_\Theta b$. If the first is the case, then also $a : \Diamond b'$ occurs at Θ, and $b' \in W$, so $[a]_\approx \overline{R}[b']_\approx$, and trivially, $[b']_\approx = [b]_\approx$. If the second is the case, and moreover, $a' = a$ and $b' = b$, then clearly $[a]_\approx \overline{R}[b]_\approx$. If $a' = a$ and $b' \neq b$, then by Theorem 4.4, there is a nominal c of the root formula such that $b' \sim_\Theta c$. But then also $a : \Diamond c$ occurs at Θ, and $c \in W$, so $[c]_\approx \overline{R}[c]_\approx$, and trivially, $[c]_\approx = [b]_\approx$. If $a' \neq a$ and $b' = b$, then by Theorem 4.4, there is a nominal c of the root formula such that $a' \sim_\Theta c$. But then also $c : \Diamond b$ occurs at Θ, and $c \in W$, so $[a]_\approx \overline{R}[b]_\approx$, and trivially, $[c]_\approx = [a]_\approx$. If $a' \neq a$ and $b' \neq b$, then by Theorem 4.4, there are nominals c and d of the root formula such that $a' \sim_\Theta c$ and $b' \sim_\Theta d$. But then also $c : \Diamond d$ occurs at Θ, and $c, d \in W$, so $[c]_\approx \overline{R}[d]_\approx$, and trivially, $[c]_\approx = [a]_\approx$ and $[d]_\approx = [b]_\approx$. We now consider the case where ψ is a not nominal. By the rule $(\Diamond)$ also some formulas $a : \Diamond c$ and $c : \psi$ occur at Θ where the nominal c is new (note that $a \in W$, so the application of the rule is not blocked by a loop-check condition). If $c \in W$, then clearly $[a]_\approx \overline{R}[c]_\approx$, and by induction $\mathcal{M}, g, [c]_\approx \models \psi$. If $c \notin W$, then by definition of W there exists a nominal d such that $c \subseteq_\Theta d$. Without loss of generality we assume that there does not exist a nominal e such that $d \subseteq_\Theta e$. But this implies that $d \in W$. Moreover, by Lemma 4.1, the formula ψ is a subformula of the root formula, so $d : \psi$ occurs at Θ. By induction, $\mathcal{M}, g, [d]_\approx \models \psi$, and clearly, $[a]_\approx \overline{R}[d]_\approx$.

Assume that $\neg a : \Diamond\psi$ occurs at Θ. We then have to prove that $\mathcal{M}, g, [a]_\approx \models \Diamond\psi$ does not hold, that is, for any equivalence class $[c]_\approx$ such that $[a]_\approx \overline{R}[c]_\approx$, it is not the case that $\mathcal{M}, g, [c]_\approx \models \psi$. From $[a]_\approx \overline{R}[c]_\approx$ it follows that there exists nominals $a' \approx a$ and $c' \approx c$, satisfying one of the three conditions in the definition of the relation R. In the first and third condition in this definition, the formula $a' : \Diamond c'$ occurs at Θ. Then $a : \Diamond c$ also occurs at Θ, and by the rule $(\neg\Diamond)$ also $\neg c : \psi$. By induction we conclude that $\mathcal{M}, g, [c]_\approx \models \psi$ does not hold. In the second condition in the definition, there exists a nominal d in N^Θ such that the formula $a' : \Diamond d$ occurs at Θ and $d \subseteq_\Theta c'$. Then $a : \Diamond d$ also occurs at Θ, and by the rule $(\neg\Diamond)$ also $\neg d : \psi$. But by Lemma 4.1, the formula ψ is a subformula of the root formula, and $d \subseteq_\Theta c'$, so $\neg c' : \psi$ occurs at Θ. By induction we conclude that $\mathcal{M}, g, [c']_\approx \models \psi$ does not hold and trivially, $[c']_\approx = [c]_\approx$. □

5 Related work

In ordinary modal logic, loop-checks are used in connection with standard Fitting-style prefixed tableau systems for transitive logics such as K4, see [6] and [10]. Early applications of loop-checks can be found in [9] and [5]. Now, a simple prefixed tableau system can be formulated for the modal logic K such that a systematic tableau construction always terminates. The systematic tableau construction algorithm for K does not involve loop-checks. However, a systematic tableau construction may not terminate if the tableau system for K is extended with the standard prefixed tableau rule

$$\frac{s : \Box\phi, s < t}{t : \Box\phi}$$

for transitivity (the notation should be self-explanatory), whereby a tableau system for K4 is obtained. Intuitively, the problem is that the rule allows information to be moved forward from a world to any accessible world. The standard way to fix this problem is to incorporate loop-check conditions on the applications of existential rules. The intuitive reason why this technique works in the context of hybrid logic too, is that the problem here also is that information can be moved between worlds, namely in connection with applications of the rule (Id) in the Tzakova-style system and similarly, in connection with the rules $(Nom1)$ and $(Nom2)$ in the Blackburn-style system. Intuitively, these rules allow information to be moved between worlds that are identical.

Nominals are often used in description logics, and certain tableau-based decision procedures for such logics also make use of loop-checks. This is for example the case with the decision procedure given in [7] which is based on a prefixed tableau system in line with our Tzakova-style system.[2] The logic given in that paper, and other similar logics, do not involve satisfaction operators or the universal modality, but it is well-known that if a description logic has transitive roles together with role hierarchies, which is the case with the logic in [7], then general concept inclusion axioms can be internalised into concepts, as described in [8], pages 164 and 165. This technique can also be used to define an "approximation" of the universal modality: Given roles $R_1, \ldots, R_n$ occurring in a formula ϕ and a new role U, a set of role axioms

$$\{\mathsf{Trans}(U), R_1 \sqsubseteq U, ..., R_n \sqsubseteq U\}$$

is defined ensuring that the relation for the role U is transitive and contains the relations for all the other roles. It follows that ϕ is satisfiable wrt. arbitrary models if and only if the formula ϕ' obtained by replacing any universal modality $E\psi$ in ϕ by $\psi \wedge \forall U.\psi$ is satisfiable wrt. models satisfying the axioms. In the terminology of modal logic, this is the case since the axioms ensures that $\psi \wedge \forall U.\psi$ is true at a world w if and only if ψ is true at a set of worlds containing the submodel generated by w. In case nominals are involved, further axioms have to be added such that $\psi \wedge \forall U.\psi$ is true at a world w if and only if ψ is true at a set of worlds containing the submodel generated by the set of worlds consisting of w together with the denotations of all nominals in ψ. In this sense, the universal modality can be approximated if further machinery is present, namely axioms involving transitive roles and role hierarchies.

However, we think that the universal modality and satisfaction operators are so important and widely used that it justifies independent and direct tableau-based decision procedures, as given in the present paper. Also, it seems unnecessarily complicated to obtain a decision procedure encompassing the universal modality (which is first-order definable) by a reduction to a decision procedure involving axioms for a new role (which amounts to imposing a second-order condition on models, namely the condition that there exists a relation satisfying the axioms).

Acknowledgements: Thanks to Patrick Blackburn for comments on the work presented here, in particular for suggesting to incorporate the universal modality. Also thanks to Jørgen Villadsen for comments. The authors are partially supported by the Danish Natural Science Research Council in connection with the HyLoMOL project.

[2] We thank one of the anonymous referees for pointing this out.

References

[1] Areces, C., P. Blackburn and M. Marx, *Hybrid logics: Characterization, interpolation and complexity*, Journal of Symbolic Logic **66** (2001), pp. 977–1010.

[2] Blackburn, P., *Internalizing labelled deduction*, Journal of Logic and Computation **10** (2000), pp. 137–168.

[3] Blackburn, P., M. de Rijke and Y. Venema, "Modal Logic," Cambridge Tracts in Theoretical Computer Science **53**, Cambridge University Press, 2001.

[4] D'Agostino, M., D. Gabbay, R. Hähnle and J. Posegga, editors, "Handbook of Tableau Methods," Springer, 1999.

[5] Fitting, M., "Proof Methods for Modal and Intuitionistic Logic," Reidel, 1983.

[6] Goré, R., *Chapter 6: Tableau methods for modal and temporal logics*, in: *Handbook of Tableau Methods*, Kluwer Academic Publishers, 1999 pp. 297–396.

[7] Horrocks, I. and U. Sattler, *Ontology reasoning in the* $\mathcal{SHOQ}(D)$ *description logic*, in: *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001)*, 2001, pp. 199–204.

[8] Horrocks, I., U. Sattler and S. Tobies, *Practical reasoning for expressive description logics*, in: *Proceedings of LPAR'99*, Lecture Notes in Artificial Intelligence **1705** (1999), pp. 161–180.

[9] Hughes, G. and M. Cresswell, "An Introduction to Modal Logic," Methuen, 1968.

[10] Massacci, F., *Single step tableaux for modal logics*, Journal of Automated Reasoning **24** (2000), pp. 319–364.

[11] Tzakova, M., *Tableaux calculi for hybrid logics*, in: N. V. Murray, editor, *Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX 1999*, Lecture Notes in Artificial Intelligence **1617** (1999), pp. 278–292.