TABLEAU-BASED DECISION PROCEDURES FOR HYBRID LOGIC

THOMAS BOLANDER AND TORBEN BRAÜNER

Abstract. Hybrid logics are a principled generalization of both modal logics and description logics. It is well-known that various hybrid logics without binders are decidable, but decision procedures are usually not based on tableau systems, a kind of formal proof procedure that lends itself towards computer implementation. In this paper we give four different tableau-based decision procedures for a very expressive hybrid logic including the universal modality; three of the procedures are based on different tableau systems, and one procedure is based on a Gentzen system. 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. The loop-checks used in our four decision procedures are similar, but the four proof systems on which the procedures are based constitute a spectrum of different systems: prefixed and internalized systems, tableau and Gentzen systems.

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

This is a pre-print. The final version of the paper will appear in Journal of Logic and Computation.

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 [14]. 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. We also consider a variant of a tableau system given by van Eijck in the paper [15]. For this system we also provide a terminating tableau construction algoritm, along the same lines as the algorithm provided for the system of Tzakova. Furthermore, we 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. Finally, we consider a reformulation of Blackburn's system as a Gentzen calculus and discuss how to reformulate the decision procedure. Analogous results follow 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 our variant of van Eijck's tableau system. In the fifth section we give the decision procedure for Blackburn's tableau system, and in section 6 we reformulate this system as a Gentzen sequent system. In the final section we discuss some related work. This paper is a revised and extended version of a workshop paper which appeared as [4].

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*

(1) W *is a non-empty set;*

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

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

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

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

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

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

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

FIGURE 1. Modified version of Tzakova's tableau rules

(2) *R is a binary relation on W; and*
(3) *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.

$$\begin{array}{rll}
\mathcal{M},g,w\models p & \text{iff} & V(w,p)=1\\
\mathcal{M},g,w\models a & \text{iff} & w=g(a)\\
\mathcal{M},g,w\models\neg\phi & \text{iff} & \text{not } \mathcal{M},g,w\models\phi\\
\mathcal{M},g,w\models\phi\wedge\psi & \text{iff} & \mathcal{M},g,w\models\phi \text{ and } \mathcal{M},g,w\models\psi\\
\mathcal{M},g,w\models a:\phi & \text{iff} & \mathcal{M},g,g(a)\models\phi\\
\mathcal{M},g,w\models\Diamond\phi & \text{iff} & \text{for some } v\in W,\ wRv \text{ and } \mathcal{M},g,v\models\phi\\
\mathcal{M},g,w\models E\phi & \text{iff} & \text{for some } v\in W,\ \mathcal{M},g,v\models\phi
\end{array}$$

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 [14] is a prefixed tableau calculus (see the book [5] 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, and a $(\neg)$ rule. The $(\neg)$ rule can be dropped, but that would give a slightly less transparent model construction in the completeness proof. Even though our

calculus differs from Tzakova's in these ways, we will still refer to ours as *Tzakova's system*. A *tableau* in Tzakova's system is a well-founded tree in which each node is labelled with a prefixed formula or an accessibility formula, and the edges represent applications of tableau rules in the usual way. The rules $(\neg)$, $(:)$, $(\neg:)$, $(\Diamond)$, and (E) are called *prefix generating rules*. Whenever one of these rules is applied to a branch, a new prefix will be introduced to the branch. We impose the following conventions on the application of rules in tableau constructions.

- In constructing a tableau, no prefix generating rule is ever applied to the same premise twice on the same branch.
- A formula is never added to a tableau branch where it already occurs.

Later we will show how to construct a model from an open tableau branch in Tzakova's system. The set of worlds in such a model is chosen as a subset of the prefixes occurring on the branch, and if $\sigma\phi$ occurs on the branch ϕ will be true in the world σ. Thus, intuitively, one can think of the prefixes as worlds and prefixed formulas $\sigma\phi$ occurring on branches as expressing: "ϕ is true at σ". Similarly, accessibility formulas $\sigma < \sigma'$ can intuitively be thought of as expressing: "the world σ' is accessible from the world σ".

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 prefix generating rules, $(\neg)$, $(:)$, $(\neg:)$, $(\Diamond)$ 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 prefix generating rules 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 well-founded, finitely branching tree.*

Proof. That the graph is well-founded follows from the observation that if $\sigma <_\Theta \tau$, then the first occurrence of σ on Θ is before the first occurrence of τ. 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 prefix generating rules). □

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. The* urfather *of a prefix σ on Θ is the earliest occurring prefix on Θ which σ is included in. The* urfather *of σ on Θ is denoted $u_\Theta(\sigma)$. Prefixes σ on Θ for which $u_\Theta(\sigma) = \sigma$ are called* urfathers *on Θ.*

Note that if $\sigma = u_\Theta(\tau)$ for some prefix τ then $u_\Theta(\sigma) = \sigma$. In other words, if σ is an urfather *of* a prefix τ on a branch Θ, then σ is an urfather on Θ. 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 an arbitrary rule to $\mathcal{T}_i$ subject to the following restriction:*

($\mathcal{R}$) *A prefix generating rule is only allowed to be applied to a formula $\sigma\phi$ on a branch of $\mathcal{T}_i$ if σ is an urfather on that branch.*

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

Restriction $\mathcal{R}$ is our loop-check condition. Intuitively the condition says that we are not allowed to construct a new world σ' from an existing world σ if there is an earlier introduced world τ in which everything true at σ is also true. In other words: In order to be allowed to construct a new world σ' from an existing world σ, the world σ needs to contain some additional information compared to the earlier introduced worlds.

Theorem 3.5 (**Termination**). *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 branch Θ. By the tableau conventions, all prefixed formulas along this branch 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 well-founded, 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 σ_j is included in σ_i with respect to Θ_j. Let thus 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 σ_j is included in σ_i with respect to Θ_j. From this it follows that σ_j can not be an urfather on Θ_j, since σ_i has its first occurrence on Θ_j before σ_j. 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 prefix generating rules to a formula $\sigma_j\psi$ at Θ_j. However, this is in contradiction with restriction $\mathcal{R}$ by which none of the prefix generating rules can be applied to the formula $\sigma_j\psi$ at Θ_j since σ_j is not an urfather on that branch. □

Example 3.6. Consider the hybrid formula $c \wedge \Diamond c$, where c is a nominal. Without the loop-check condition $\mathcal{R}$, an infinite tableau with root $\sigma(c\wedge\Diamond c)$ can easily be constructed, as shown in Figure 2. Note that in this infinite tableau we keep on constructing the same world—the world referred to by c—over and over again. We just give new prefixes to name the world each time it is reconstructed: $\sigma', \sigma'', \sigma''', \ldots$. If we apply restriction $\mathcal{R}$ then the second application of the ($\Diamond$) rule on the branch will be blocked, since at the time it is applied σ' is not an urfather—the urfather of σ' is σ. Thus with restriction $\mathcal{R}$ in play the tableau can not become infinite. The restriction blocks constructing the same world over and over again, since by the restriction a new world is not allowed to be constructed from a world if there exists an earlier introduced copy of it.

3.3. **Soundness and completeness.** Soundness of the tableau calculus in Figure 1 can be proved by showing that each rule preserves satisfiability [14]. 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

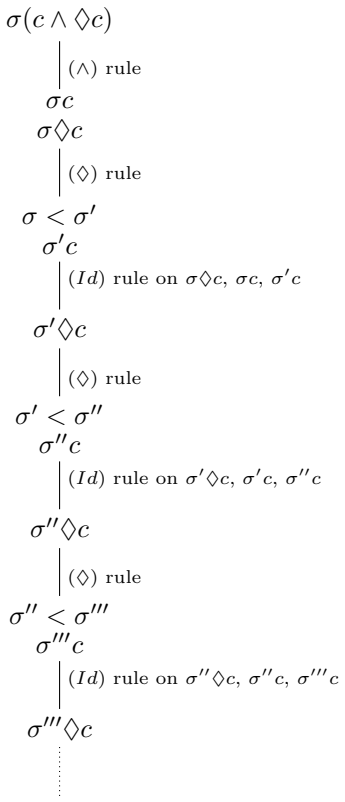

FIGURE 2. An infinite tableau without restriction $\mathcal{R}$.

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 couple of simple result.

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 prefix generating rules $(\neg)$, $(:)$, $(\neg:)$, $(\Diamond)$ and (E) whenever the premise is a formula occurrence $\sigma\phi$ 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 prefix generating rules, we immediately get that the tableau is closed under all rules except possibly these. Now consider the prefix generating 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, the rule $(\Diamond)$ is not blocked by restriction $\mathcal{R}$ on $\mathcal{T}_n$. 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 other prefix generating rules is proved similarly. □

Lemma 3.8. *Let Θ be a branch of a tableau and let σ and τ be prefixes occurring on Θ. Suppose there exists a nominal c such that both σc and τc occurs on Θ. Then for all formulas ϕ, $\sigma\phi$ occurs on Θ if and only if $\tau\phi$ occurs on Θ.*

Proof. By symmetry, we only have to prove that if $\sigma\phi$ is a prefixed formula occurring on Θ then $\tau\phi$ occurs on Θ as well. Let thus $\sigma\phi$ be a prefixed formula occurring on Θ. That $\tau\phi$ occurs on Θ as well now 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 with root $\sigma_0\phi_0$, we define the model $\mathcal{M}_\Theta$ and a corresponding assignment g_Θ by

$$\begin{aligned}
\mathcal{M}_\Theta &= (W_\Theta, R_\Theta, V_\Theta), \text{ where} \\
W_\Theta &= \{u_\Theta(\sigma) \mid \sigma \text{ occurs on } \Theta\} \\
R_\Theta &= \{(\sigma, u_\Theta(\tau)) \in W_\Theta^2 \mid \sigma < \tau \text{ occurs on } \Theta\} \\
V_\Theta(\sigma, p) &= 1 \text{ iff } \sigma p \text{ occurs on } \Theta. \\
g_\Theta(c) &= \begin{cases} \sigma_0 & \text{if there is no } \sigma \text{ for which } \sigma c \text{ occurs on } \Theta \\ u_\Theta(\sigma) & \text{if } \sigma c \text{ occurs on } \Theta \end{cases}
\end{aligned}$$

We need to check that g_Θ is a well-defined assignment for $\mathcal{M}_\Theta$. First of all, we note that the prefix σ_0 of the root formula is always an urfather. Furthermore, note that if σ and σ' are prefixes such that both σc and $\sigma' c$ occur on Θ, then it follows from Lemma 3.8 that $u_\Theta(\sigma) = u_\Theta(\sigma')$. This proves g_Θ to be a well-defined assignment. 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 $\sigma\neg p$ occurs on Θ where p is a propositional symbol and σ is an urfather. Then σp does not occur on Θ, since Θ is an open branch. We therefore get $V_\Theta(\sigma, p) = 0$ which implies $\mathcal{M}_\Theta, g_\Theta, \sigma \models \neg p$. 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. Assume now $\sigma\neg c$ occurs on Θ where c is a nominal and σ is an urfather. Then by closure under the rule $(\neg)$ (Lemma 3.7) we get that τc occurs on Θ for some prefix τ. This implies $g_\Theta(c) = \tau$. Since Θ is an open branch, σc can not occur on it, and thus we get $\sigma \neq \tau$. This implies $g_\Theta(c) \neq \sigma$, and thus $\mathcal{M}_\Theta, \sigma \models \neg c$. 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 $\mathcal{M}_\Theta, g_\Theta, \sigma \models E\psi$.

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

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

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

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

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

$$\frac{\Gamma \cup \{\sigma c, \tau c\}}{\Gamma[\sigma/\tau] \cup \{\sigma c\}}\,(sub)^\star$$

$*$ The prefix σ' is new to the entire tableau.
$\dagger$ The prefix σ'' occurs in Γ.
$\star$ The prefix σ is introduced earlier on the branch than τ. $\Gamma[\sigma/\tau]$ denotes the result of substituting σ for τ everywhere in the formulas of Γ.

FIGURE 3. Rules for the substitution-based tableau calculus.

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 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. □

4. A substitution-based prefixed tableau calculus

In this section we consider a variant of the tableau calculus of van Eijck [15]. The system of van Eijck is most closely related to that of Tzakova, but instead of the (Id) rule van Eijck has a rule for nominal substitution. Rules for nominal substitution in hybrid logic have also appeared earlier in sequent calculi [12]. In its original formulation the substitution rule of van Eijck looks like this

$$\frac{\mathbf{B}, a : b}{\mathbf{B}^t_s} \quad s = \min(a, b), t = \max(a, b).$$

Here $\mathbf{B}$ is a tableau branch and $\mathbf{B}^t_s$ denotes the result of substituting the nominal s for the nominal t everywhere in the branch. The nominals s and t are the smallest and the largest, respectively, of the nominals a and b according to some fixed linear order on the nominals. Since this is a rule for replacing an entire branch by another branch the tableau calculus is implicitly working with sets of formulas at each node of a tableau rather than with individual formulas. In Figure 3 a variant of van Eijck's system is presented. The given system differs from the system presented in [15] in a number of ways. First of all, we make it explicit that the tableau rules are working on sets of formulas, so that the premises and conclusions of all rules are sets of formulas on the form $\Gamma \cup \{\cdots\}$. This gives the calculus some resemblances to the Gentzen calculus to be considered in Section 6 below. Furthermore, the system presented in Figure 3 is a *prefixed* tableau calculus like Tzakova's system, whereas the original system of van Eijck is "semi-internalized": He uses nominals instead of prefixes, but accessibility formulas are still metalinguistic expressions on the

form $a < b$ rather than formulas $a : \Diamond b$ of the object language as in Blackburn's internalized system which will be presented in Section 5 below. Thus the style of the original system of van Eijck places it in between the systems of Tzakova and Blackburn: Satisfaction statements are formulas $a : \phi$ of the object language (as in Blackburn) but accessibility formulas are expressions $a < b$ of the metalanguage (as in Tzakova). The semi-internalized nature of van Eijck's original system necessitates special rules to get from object-language formulas like $a : \Diamond b$ to meta-language formulas like $a < b$. These rules are not needed in our prefixed calculus. Another difference between our presented variant and the original system of van Eijck is that van Eijck uses a multi-modal logic and includes inverse modalities. We do not do that, but instead we have extended the system with the universal modality. We will refer to the tableau system presented in Figure 3 as the *substitution-based system*. For each rule of the system we have chosen to let all of the formulas occurring in the premise of the rule also occur in the corresponding conclusions of the rule. This is not strictly necessary to ensure completeness, but in our case it gives a simpler closure condition and simpler completeness proof.

In the rules of the substitution-based system, Γ represents an arbitrary finite set of prefixed formulas and accessibility formulas, and $\Gamma[\sigma/\tau]$ denotes the result of substituting the prefix σ for the prefix τ everywhere in the formulas of Γ. A *tableau* in the calculus is a well-founded tree in which each node is labelled with a set of formulas. If a node x has children $y_1, \ldots, y_n$ then there is an instance of a tableau rule such that x is labelled with the premise set of that rule instance and $y_1, \ldots, y_n$ are labelled with the conclusion sets (note that n will always be 1 or 2). When it will not lead to ambiguities, we will allow ourselves to identify nodes with the sets of formulas they are labelled with. Thus, for instance, if Γ is a node of a tableau, we will write $\sigma\phi \in \Gamma$ to mean that $\sigma\phi$ is among the formulas that Γ is labelled with. A branch of a tableau is said to be *closed* if it contains a node labelled by a set of formulas on the form $\Gamma \cup \{\sigma\phi, \sigma\neg\phi\}$. A branch is *open* if it is not closed. A tableau is *closed* if all branches of the tableau are closed—otherwise it is *open*. A *tableau proof* of a formula ϕ is a closed tableau with root $\{\sigma\neg\phi\}$. For each of the rules of Figure 3, the formulas shown explicitly in the premise set are called the *principal premises*. For instance, if the rule (:) is applied to a premise on the form $\Gamma \cup \{\sigma c : \phi\}$ to obtain the conclusion $\Gamma \cup \{\sigma c : \phi, \sigma' c, \sigma'\phi\}$ then the formula $\sigma c : \phi$ is called the principal premise of the application. In the rule (*sub*) we will call the principal premise σc the *first principal premise* and the premise τc the *second principal premise*. As for Tzakova's system the rules ($\neg$), (:), ($\neg$:), ($\Diamond$) and (E) will be called *prefix generating rules*. We impose similar conventions on the application of tableau rules as for Tzakova's system. These conventions are denoted $\mathcal{C}_1$ and $\mathcal{C}_2$ and are defined by:

- ($\mathcal{C}_1$) In constructing a tableau, no prefix generating rule is ever applied to the same set of principal premises twice on the same branch.
- ($\mathcal{C}_2$) A rule instance is never applied to a premise set Γ if the conclusion set Γ' of the instance is identical to Γ.

Example 4.1. Consider again the formula $c \wedge \Diamond c$ introduced in Example 3.6. Figure 4 shows a finite tableau in the substitution-based calculus with root $\{\sigma(c \wedge \Diamond c)\}$. Compare it with the tableau in Tzakova's system given in Figure 2. The present tableau is a finite open branch, and it is furthermore saturated: No rule applies to the leaf satisfying the rule application conventions $\mathcal{C}_1$ and $\mathcal{C}_2$. Thus with rule (Id) replaced by (*sub*) we don't need loop-checking to ensure termination of the tableau with root formula $c \wedge \Diamond c$. However, the presence of the universal modality in the calculus still makes it necessary to have some kind of loop-checking to ensure termination of tableau construction in generel. This is illustrated by the infinite tableau in Figure 5. To improve readability of the tableau, all nodes except the root are labelled with a set of formulas written on the form $\Gamma \cup \Gamma'$ where Γ' is the set of conclusions of the rule application leading to the node. Since this tableau is infinite, replacing (Id) by (*sub*) is not sufficient to allow us to prove termination without using loop-checks.

Termination, soundness and completeness of the substitution-based system is going to be proved by relating it to Tzakova's system.

$$\{\sigma(c \wedge \Diamond c)\}$$
$$\Big|\ (\wedge)\text{ rule}$$
$$\{\sigma(c \wedge \Diamond c), \sigma c, \sigma\Diamond c\}$$
$$\Big|\ (\Diamond)\text{ rule}$$
$$\{\sigma(c \wedge \Diamond c), \sigma c, \sigma\Diamond c, \sigma < \sigma', \sigma' c\}$$
$$\Big|\ (sub)\text{ rule: substitute } \sigma \text{ for } \sigma'$$
$$\{\sigma(c \wedge \Diamond c), \sigma c, \sigma\Diamond c, \sigma < \sigma\}$$

FIGURE 4. A tableau in the substitution-based calculus.

$$\{\sigma\neg E\neg\Diamond p\}$$
$$\Big|\ (\neg E)\text{ rule}$$
$$\{\sigma\neg E\neg\Diamond p\} \cup \{\sigma\neg\neg\Diamond p\}$$
$$\Big|\ (\neg\neg)\text{ rule}$$
$$\{\sigma\neg E\neg\Diamond p, \sigma\neg\neg\Diamond p\} \cup \{\sigma\Diamond p\}$$
$$\Big|\ (\Diamond)\text{ rule}$$
$$\{\sigma\neg E\neg\Diamond p, \sigma\neg\neg\Diamond p, \sigma\Diamond p\} \cup \{\sigma < \sigma', \sigma' p\}$$
$$\Big|\ (\neg E)\text{ rule}$$
$$\{\sigma\neg E\neg\Diamond p, \sigma\neg\neg\Diamond p, \sigma\Diamond p, \sigma < \sigma', \sigma' p\} \cup \{\sigma'\neg\neg\Diamond p\}$$
$$\Big|\ (\neg\neg)\text{ rule}$$
$$\{\sigma\neg E\neg\Diamond p, \sigma\neg\neg\Diamond p, \sigma\Diamond p, \sigma < \sigma', \sigma' p, \sigma'\neg\neg\Diamond p\} \cup \{\sigma'\Diamond p\}$$
$$\Big|\ (\Diamond)\text{ rule}$$
$$\{\sigma\neg E\neg\Diamond p, \sigma\neg\neg\Diamond p, \sigma\Diamond p, \sigma' p, \sigma'\neg\neg\Diamond p, \sigma'\Diamond p\} \cup \{\sigma' < \sigma'', \sigma'' p\}$$
$$\Big|\ (\neg E)\text{ rule}$$
$$\{\sigma\neg E\neg\Diamond p, \sigma\neg\neg\Diamond p, \sigma\Diamond p, \sigma' p, \sigma'\neg\neg\Diamond p, \sigma'\Diamond p, \sigma' < \sigma'', \sigma'' p\} \cup \{\sigma''\neg\neg\Diamond p\}$$
$$\vdots$$

FIGURE 5. An infinite tableau in the substitution-based calculus.

4.1. **Systematic tableau construction.** Everything from the systematic tableau construction and termination proof of Tzakova's system carries over to the substitution-based system with only minor changes. Of course the substitution-based system satisfies the quasi-subformula property (Lemma 3.1). We just have to note that when saying that a formula $\sigma\phi$ *occurs* in a tableau in the substitution-based system we mean that the tableau contains a node of the form $\Gamma \cup \{\sigma\phi\}$. Similarly, we say that a formula $\sigma\phi$ *occurs* on a tableau branch Θ if one of the nodes of the branch has the form $\Gamma \cup \{\sigma\phi\}$. We can again define $\sigma <_\Theta \sigma'$ to hold if σ' is a prefix introduced to the branch Θ by an application of one of the prefix generating rules to a principal premise on the form $\sigma\phi$. When $\sigma <_\Theta \sigma'$ we say that σ' is *generated* by σ on Θ. It should be noted that even if $\sigma <_\Theta \sigma'$ holds it does not necessarily imply that σ and σ' are prefixes occurring in the leaf of Θ. The prefixes might have been replaced by other prefixes using the (*sub*) rule on the branch. This does not affect the definition of the $<_\Theta$ relation, however, and Proposition 3.2 still holds with the proof unchanged. We can define *inclusion*, *urfathers* and the map u_Θ exactly as in Definition 3.3. The

tableau construction algorithm and the termination proof then precisely mimic the construction used for Tzakova's system, as we show below.

Definition 4.2 (**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 singleton set $\{\sigma\neg\phi\}$, 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 an arbitrary rule to the leaf of an arbitrary branch of $\mathcal{T}_i$ subject to the following restriction:*

($\mathcal{R}$) *A prefix generating rule is only allowed to be applied to a principal premise $\sigma\phi$ at the leaf of a branch of $\mathcal{T}_i$ if σ is an urfather on that branch.*

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

Theorem 4.3 (**Termination**). *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 branch Θ. We will show that infinitely many different prefixes must occur on Θ. Assume to obtain a contradiction that Θ only contains finitely many different prefixes. Then the (*sub*) rule can only have been applied finitely many times on Θ. Thus there must exist an infinite final segment Θ' of Θ on which (*sub*) has not been applied. Looking at the rules of the calculus, this implies that if Γ, Γ' are consecutive nodes on Θ then the set of formulas Γ' must have larger cardinality than the set of formulas Γ. Thus the set of all formulas occurring on Θ' must be infinite. By the quasi-subformula this implies that Θ' contains infinitely many different prefixes. Since Θ' is a final segment of Θ this gives a contradiction. Thus we have proven that Θ contains infinitely many different prefixes. This implies that the graph $(N^\Theta, <_\Theta)$ is infinite. Since by Proposition 3.2 the graph is a well-founded, 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 node 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 (quasi-subformula property). 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 σ_j is included in σ_i on Γ_j. Let thus ϕ be an arbitrary formula for which $\sigma_j\phi$ occurs on Θ_j. Then $\phi \in \Gamma_j$ and since $\Gamma_j = \Gamma_i$ we get that $\sigma_i\phi$ occurs on Θ_i. Since Θ_i is an initial segment of Θ_j this implies that $\sigma_i\phi$ occurs on Θ_j. This proves that σ_j is included in σ_i on Θ_j. Note that furthermore σ_i has its first occurrence on Θ_j before σ_j, since $i < j$. Thus σ_j can not be an urfather on Θ_j. Now consider the first node containing an occurrence of σ_{j+1}. By definition of Θ_j, this node is the child of the last node of Θ_j. The prefix σ_{j+1} is generated by σ_j, so σ_{j+1} must be introduced by applying a prefix generating rule to a principal premise of the form $\sigma_j\psi$ on Θ_j. However, this is in contradiction with restriction $\mathcal{R}$, since σ_j is not an urfather on Θ_j. □

According to van Eijck in [15], his tableau calculus can be made into a decision procedure for the logic. However, he only gives a very brief sketch of a termination proof, and it is based on a rather complicated proof procedure which deviates quite significantly from the pure tableau calculus itself. Our proof procedure for the substitution-based system is much more directly based on the tableau calculus with only a single condition, restriction $\mathcal{R}$, to ensure termination.

4.2. **Soundness and completeness.** Soundness of the substitution-based system is simple to prove. Except for the (*sub*) rule all the rules are proven to preserve satisfiability exactly as for Tzakova's system. To prove that (*sub*) preserves satisfiability we simply have to note that if both the world σ and the world τ are referred to by the nominal c, then σ and τ must be the same world. We now turn to the completeness proof. It is possible to prove completeness by constructing a translation mapping from tableau branches of the substitution-based system into branches of Tzakova's system. However, since such a mapping becomes rather complex, it appears to be simpler to prove completeness directly through model construction as for Tzakova's system. First we need a couple of new lemmata. Given a tableau branch Θ we denote its leaf by L_Θ.

Lemma 4.4. *Let Θ be a branch of a tableau constructed according to the substitution-based tableau construction algorithm. If $\sigma c, \sigma' c \in L_\Theta$ then $\sigma = \sigma'$.*

Proof. Suppose $\sigma c, \sigma' c \in L_\Theta$. Since L_Θ is the leaf of a tableau constructed according to the tableau construction algorithm, no rule satisfying restriction $\mathcal{R}$ and conventions $\mathcal{C}_1, \mathcal{C}_2$ can be applied to L_Θ. In particular, (*sub*) can not be applied to the principal premises $\sigma c, \sigma' c$ in a way that satisfies convention $\mathcal{C}_2$. The only reason there can be for this is that $\sigma = \sigma'$. □

Lemma 4.5. *Let Θ be a branch of a tableau in the substitution-based system, and let Γ and Δ be nodes in Θ such that Δ is a descendant of Γ. If $\sigma\phi \in \Gamma$ and σ is a prefix occurring in Δ then $\sigma\phi \in \Delta$.*

Proof. Let Θ' denote the subpath of Θ with initial node Γ and final node Δ. Note that if a rule other than (*sub*) is applied to a premise containing $\sigma\phi$ then the conclusion must also contain $\sigma\phi$. The same holds if the rule is (*sub*) with second principal premise τc for some $\tau \neq \sigma$. Thus to prove the lemma we only need to prove that on Θ' the rule (*sub*) has not been applied with second principal premise on the form σc. So assume to obtain a contradiction that (*sub*) has been applied with second principal premise σc somewhere on Θ'. In this case all occurrences of σ has somewhere on Θ' been replaced by some other prefix. This implies that σ can not occur in the final node Δ of Θ', which contradicts the assumption on Δ. □

Lemma 4.6. *Let Θ be a branch of a tableau in the substitution-based system and let Γ and Δ be nodes in Θ such that Δ is a descendant of Γ. Let $\sigma_1, \sigma_2, \ldots, \sigma_n$ denote the prefixes occurring in Θ that are* not *urfathers on Θ. There exist prefixes $\sigma'_1, \sigma'_2, \ldots, \sigma'_n$ such that*

$$\Gamma[\sigma'_1/\sigma_1, \sigma'_2/\sigma_2, \ldots, \sigma'_n/\sigma_n] \subseteq \Delta.$$

Proof. Let Θ' denote the subpath of Θ with initial node Γ and final node Δ. The proof is by induction on the length of Θ'. If the length of Θ' is 0 we have $\Delta = \Gamma$ and the result is trivial. Assume now that the result is proven for paths of length up to m and assume that Θ' has length $m + 1$. Let Δ' denote the parent node of Δ on Θ. By induction hypothesis, there exist prefixes $\sigma'_1, \ldots, \sigma'_n$ such that

$$\Gamma[\sigma'_1/\sigma_1, \ldots, \sigma'_n/\sigma_n] \subseteq \Delta'. \tag{1}$$

If the rule applied to get Δ from Δ' is any other than (*sub*) then we have $\Delta' \subseteq \Delta$ and thus, by (1),

$$\Gamma[\sigma'_1/\sigma_1, \ldots, \sigma'_n/\sigma_n] \subseteq \Delta$$

as needed. If the rule applied to get Δ from Δ' is (*sub*) with second principal premise σc for some nominal c then we get $\Delta = \Delta'[\sigma'/\sigma]$ for some prefix σ' introduced earlier to Θ than σ. This implies

$$\Gamma[\sigma'_1/\sigma_1, \ldots, \sigma'_n/\sigma_n][\sigma'/\sigma] \subseteq \Delta'[\sigma'/\sigma] = \Delta,$$

using (1). If we can prove the existence of prefixes $\sigma''_1, \ldots, \sigma''_n$ such that

$$\Gamma[\sigma'_1/\sigma_1, \ldots, \sigma'_n/\sigma_n][\sigma'/\sigma] = \Gamma[\sigma''_1/\sigma_1, \ldots, \sigma''_n/\sigma_n] \tag{2}$$

then we are done. We will first prove that σ must be included in σ' on Θ. Suppose therefore that $\sigma\phi$ occurs on Θ for some formula ϕ. We need to prove that $\sigma'\phi$ occurs on Θ as well. The occurrence(s) of $\sigma\phi$ on Θ must come before Δ, since in Δ the prefix σ has been replaced by σ'. Since $\sigma\phi$ occurs before Δ on Θ and since σ occurs in Δ', Lemma 4.5 implies that $\sigma\phi$ occurs in Δ'. Because $\Delta = \Delta'[\sigma'/\sigma]$ this implies that $\sigma'\phi$ occurs in Δ. This concludes the proof that σ is included in σ'. Because σ' is furthermore introduced earlier to Θ than σ we get that σ can not be an urfather on Θ. Thus either σ is one of the prefixes $\sigma_1, \ldots, \sigma_n$ or it is a prefix not occurring in Γ. In both cases it is obvious that $\sigma''_1, \ldots, \sigma''_n$ can be chosen to make (2) hold. □

Corresponding to Lemma 3.7 we have the following closure result concerning tableaus in the substitution-based system.

Lemma 4.7. *Let Θ be a branch of a tableau constructed according to the substitution-based tableau construction algorithm. The set of formulas L_Θ is closed under each of the rules $(\neg\neg)$, $(\wedge)$, $(\neg\wedge)$, $(\neg\Diamond)$ and $(\neg E)$ of Figure 3. Furthermore, L_Θ is closed under the prefix generating rules $(\neg)$, $(:)$, $(\neg:)$, $(\Diamond)$ and (E) whenever the premise is a formula $\sigma\phi$ where σ is an urfather on Θ.*

Proof. Since Θ is constructed according to the algorithm, no rule applies to its leaf L_Θ satisfying restriction $\mathcal{R}$ and conventions $\mathcal{C}_1, \mathcal{C}_2$. In particular, none of the non-prefix generating rules $(\neg\neg)$, $(\wedge)$, $(\neg\wedge)$, $(\neg\Diamond)$ and $(\neg E)$ apply to L_Θ satisfying convention $\mathcal{C}_2$. This immediately implies closure of L_Θ under these rules. We now turn to the prefix generating rules. Consider first the prefix generating rule $(\Diamond)$. Assume $\sigma\Diamond\phi \in L_\Theta$ where σ is an urfather on Θ. We need to prove that L_Θ contains $\sigma'\phi$ and $\sigma < \sigma'$ for some prefix σ'. By definition of Θ, no rule applies to $\sigma\Diamond\phi$ that satisfies restriction $\mathcal{R}$ and convention $\mathcal{C}_1$. However, since σ is an urfather on the branch, the rule $(\Diamond)$ is not blocked by restriction $\mathcal{R}$. Thus the rule application must be blocked by $\mathcal{C}_1$. This implies that some node Γ of Θ contains formulas of the form $\sigma'\phi$ and $\sigma < \sigma'$. Since σ is an urfather, Lemma 4.6 now implies that L_Θ must contain formulas of the form $\sigma''\phi$ and $\sigma < \sigma''$ for some prefix σ''. This proves closure under $(\Diamond)$. Closure under the other prefix generating rules is proved similarly. □

We are now ready to define the models to be used in the completeness proof. Given a tableau branch Θ with root $\{\sigma_0\phi_0\}$ we define the model $\mathcal{M}_\Theta$ and corresponding assignment g_Θ by

$$\begin{aligned}
\mathcal{M}_\Theta &= (W_\Theta, R_\Theta, V_\Theta), \text{ where} \\
W_\Theta &= \{u_\Theta(\sigma) \mid \sigma \text{ occurs in } L_\Theta\} \\
R_\Theta &= \{(\sigma, u_\Theta(\tau)) \in W_\Theta^2 \mid \sigma < \tau \in L_\Theta\} \\
V_\Theta(\sigma, p) &= 1 \text{ iff } \sigma p \in L_\Theta \\
g_\Theta(c) &= \begin{cases} \sigma_0 & \text{if there is no } \sigma \text{ for which } \sigma c \in L_\Theta \\ \sigma & \text{if } \sigma c \in L_\Theta \end{cases}
\end{aligned}$$

The well-definedness of g_Θ follows immediately from Lemma 4.4. Note that the models $\mathcal{M}_\Theta$ just defined only differs from the models defined for our version of Tzakova's system by restricting the set of formulas considered to the formulas occurring at the leaf of Θ. Thus to prove completeness for the substitution-based system we can directly reuse most of the completeness proof given for our version of Tzakova's system.

For the completeness proof below, note that the root formula of a tableau in the substitution-based system will also occur at all the leafs of the tableau. Thus to prove completeness it suffices to prove satisfiability of the formulas occurring at the leaf of an open tableau branch.

Theorem 4.8 (Completeness). *Let Θ be an open branch in a tableau constructed according to the algorithm of Section 4.1. For any prefixed formula $\sigma\phi \in L_\Theta$ where σ is an urfather on Θ we have $\mathcal{M}_\Theta, g_\Theta, \sigma \models \phi$.*

Proof. We can copy the proof of Theorem 3.9 with only very minor changes. In the proof we replace all occurrences of the expression 'occurs on Θ' by 'occurs in L_Θ'. Furthermore, all references to the closure lemma, Lemma 3.7, are replaced by references to the new closure lemma, Lemma 4.7. Apart from this the proof goes through unchanged. The only extra thing we have to note is that during the proof of Theorem 3.9 we several times use the fact that if $\sigma\phi$ occurs on Θ then so does $\sigma'\phi$ where σ' is the urfather of σ. In the present proof this needs to be translated into an argument that if $\sigma\phi$ occurs in L_Θ then also $\sigma'\phi$ occurs in L_Θ. However, this follows immediately from Lemma 4.6: If $\sigma\phi$ occurs in L_Θ then $\sigma'\phi$ occurs on Θ since σ' is the urfather of σ; and by Lemma 4.6, $\sigma'\phi$ must then also occur in L_Θ since σ' is an urfather. □

5. 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 6. 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

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

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

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

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

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

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

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

$*$ 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).

FIGURE 6. Blackburn's tableau rules and rules for the universal modality

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 6. 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*. The rules are applied as follows.

(1) A 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.
(2) 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.
(3) If a formula to be added to a branch by applying a rule (destructive or non-destructive) already occurs 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.

5.1. **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 5.1. *(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 5.2. *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 5.3. *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 5.17.

Theorem 5.4. *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 5.1 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 5.5. *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 5.4. □

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 5.6. *A formula occurrence in 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 5.17.

Theorem 5.7. *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 5.1 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 5.8. *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 5.9. *Let Θ be a branch of a tableau. The graph $(N^\Theta, <_\Theta)$ is the disjoint union of a finite set of well-founded and finitely branching trees.*

Proof. That the graph is well-founded follows from the observation that if $a <_\Theta c$, then the first occurrence of a on the branch is before the first occurrence of c. That the graph is the disjoint union of a set of trees follows from well-foundedness together with the observation that if $a <_\Theta c$ and $b <_\Theta c$, then the nominals a and b are identical. That the set of trees is finite follows the observation that for any nominal c that occurs in the branch, but does not occur in the root formula, 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 $\chi_1, \chi_2, \ldots$ 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 $\chi_1, \chi_2, \ldots$ 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 5.1, 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 5.1. But there are only finitely many subformulas of the root formula, which contradicts that infinitely many different formulas occur in the the sequence $\chi_1, \chi_2, \ldots$. □

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*) and (*Nom*1), then $\sim^*_\Theta$ coincides with $\sim_\Theta$.

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

Definition 5.10. *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 5.11. *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 restriction:*

- *The existential rule $(\Diamond)$ is not applied to a formula occurrence $a:\Diamond\phi$ at a branch Θ if there exists a nominal b such that $a \subseteq_\Theta b$ (and analogously for the existential rule (E)).*

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 5.12. *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 5.9, 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 5.1 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_k 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.

5.3. **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 5.13. *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*) 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 5.14. *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.*

1. *The formula $a' : \Diamond c'$ occurs at Θ as an accessibility formula occurrence.*
2. *There exists a nominal d in N^Θ such that the formula $a' : \Diamond d$ occurs at Θ as an accessibility formula occurrence and $d \subseteq_\Theta c'$.*
3. *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 5.15. *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 (*Nom*1) 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 5.16. *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 5.17. *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 5.7, either there is a nominal b' occurring in 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 5.4, 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 $[a]_\approx \overline{R} [c]_\approx$, and trivially, $[c]_\approx = [b]_\approx$. If $a' \neq a$ and $b' = b$, then by Theorem 5.4, 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 $[c]_\approx \overline{R} [b]_\approx$, and trivially, $[c]_\approx = [a]_\approx$. If $a' \neq a$ and $b' \neq b$, then by Theorem 5.4, 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 5.1, 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 5.1, 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, $[c']_\approx = [c]_\approx$. □

6. Blackburn's tableau system reformulated as a Gentzen system

In this section we reformulate Blackburn's tableau system as a Gentzen system and we discuss how to reformulate the decision procedure. In what follows we shall sketch the basics of Gentzen systems. See [13] for further details.

Derivations in Gentzen-style have the form of finite trees where the nodes are labelled with *sequents* $\Gamma \vdash \Delta$, which are finite sets of formulas separated by the symbol $\vdash$, such that any

$$\frac{}{\Gamma, \phi \vdash \Delta, \phi}\,(Axiom)$$

$$\frac{\Gamma \vdash \Delta, a:\phi}{a:\neg\phi, \Gamma \vdash \Delta}\,(\neg L) \qquad \frac{a:\phi, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, a:\neg\phi}\,(\neg R)$$

$$\frac{a:\phi, a:\psi, \Gamma \vdash \Delta}{a:(\phi \wedge \psi), \Gamma \vdash \Delta}\,(\wedge L) \qquad \frac{\Gamma \vdash \Delta, a:\phi \quad \Gamma \vdash \Delta, a:\psi}{\Gamma \vdash \Delta, a:(\phi \wedge \psi)}\,(\wedge R)$$

$$\frac{b:\phi, \Gamma \vdash \Delta}{a:b:\phi, \Gamma \vdash \Delta}\,(:L) \qquad \frac{\Gamma \vdash \Delta, b:\phi}{\Gamma \vdash \Delta, a:b:\phi}\,(:R)$$

$$\frac{a:\Diamond c, c:\phi, \Gamma \vdash \Delta}{a:\Diamond\phi, \Gamma \vdash \Delta}\,(\Diamond L)^{**} \qquad \frac{a:\Diamond d, \Gamma \vdash \Delta, a:\Diamond\phi, d:\phi}{a:\Diamond d, \Gamma \vdash \Delta, a:\Diamond\phi}\,(\Diamond R)$$

$$\frac{c:\phi, \Gamma \vdash \Delta}{a:E\phi, \Gamma \vdash \Delta}\,(EL)^{*} \qquad \frac{\Gamma \vdash \Delta, a:E\phi, d:\phi}{\Gamma \vdash \Delta, a:E\phi}\,(ER)^{\dagger}$$

$$\frac{d:d, \Gamma \vdash \Delta}{\Gamma \vdash \Delta}\,(Ref)^{\dagger} \qquad \frac{a:\Diamond c, a:\Diamond b, b:c, \Gamma \vdash \Delta}{a:\Diamond b, b:c, \Gamma \vdash \Delta}\,(Bridge)$$

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

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

FIGURE 7. Gentzen rules for hybrid logic

sequent in a derivation is the conclusion of an instance of a Gentzen rule which has the immediate successors of the sequent in question as the premises. The root of a derivation is called the *end-sequent* of the derivation. A derivation π is called a *derivation of* a sequent $\Gamma \vdash \Delta$ if the end-sequent of π is $\Gamma \vdash \Delta$.

The intuitive reading of a sequent $\Gamma \vdash \Delta$ is that the truth of all the formulas in Γ implies the truth of at least one formula in Δ. By convention Γ, ϕ and ϕ, Γ are abbreviations for $\Gamma \cup \{\phi\}$, and similarly, Γ, Δ is an abbreviation for $\Gamma \cup \Delta$, thus, a comma on the left hand side of a sequent is intuitively read as a conjunction whereas a comma on the right hand side of a sequent intuitively is read as a disjunction.

The formulas shown explicitly in the conclusion of a rule are called *principal formulas*. Gentzen systems are characterised by having two different kinds of rules for each connective such that a Gentzen rule either introduces a connective on the left hand side of a sequent or introduces the connective on the right hand side of a sequent. Rules that introduces a connective on the left hand side of a sequent traditionally have names of the form $(\ldots L \ldots)$, and similarly, rules that introduces a connective on the right hand side of a sequent traditionally have names of the form $(\ldots R \ldots)$.

6.1. **Gentzen rules for hybrid logic.** Gentzen rules for hybrid logic are given in Figure 7. All formulas in the rules are satisfaction statements. A similar Gentzen system was considered already in [2]. Note how the Gentzen rules, except $(Axiom)$, correspond one-to-one to the tableau rules of Figure 6.

We shall make use of the following conventions about the rules, analogous to the corresponding conventions about the tableau rules. The rules $(\neg L)$, $(\neg R)$, $(\wedge L)$, $(\wedge R)$, $(:L)$, $(:R)$, $(\Diamond L)$, and

(EL) will be called *destructive* rules and the remaining rules except $(Axiom)$ will be called *non-destructive*. Note that the rules $(\Diamond R)$ and (ER) are non-destructive, and moreover, they have in-built contraction. The destructive rules $(\Diamond L)$ and (EL) will also be called *existential* rules since they introduce new nominals (rules are here read from bottom to top).

Now, the decision procedure works by searching backwards from a sequent for possible derivations of it. The search procedure finds a derivation if a derivation exists or it at some stage terminates with the information that no derivations exist, to be more precise, at the terminal stage a model and an assignment can be defined such that all the antecedent formulas are true and all the succedent formulas false. Incomplete derivations are formalized using the notion of a *pseudo-derivation* which is a well-founded tree where the nodes are labelled with sequents such that any non-leaf sequent in a pseudo-derivation is the conclusion of a rule instance which has the immediate successors of the sequent in question as the premises. A leaf sequent in a pseudo-derivation is either an arbitrary sequent or an instance of $(Axiom)$. A derivation is trivially a pseudo-derivation, and note also that a finite pseudo-derivation where any leaf sequent is an instance of $(Axiom)$, is a derivation.

6.2. **Reformulation of the decision procedure.** In the previous subsection, Blackburn's tableau rules were reformulated as Gentzen rules, which was straightforward. Note that the Gentzen rule $(Axiom)$ does not correspond to any tableau rule, rather it corresponds to a tableau branch not being open, cf. Theorem 5.17. There is an significant difference between the tableau system and the Gentzen system: When a rule is applied to a formula occurrence at a tableau branch resulting in one or two extensions of the branch, the formula occurrence is also a formula occurrence of the new branches. On the other hand, such structure sharing does not take place in Gentzen derivations, thus, we cannot directly talk about formula occurrences being identical across applications of Gentzen rules. This means that the restrictions on applications of tableau rules mentioned in the second paragraph of Section 5 have to be reformulated such that talk about formula occurrences is replaced by talk about the form of formulas:

(1) A destructive rule is not applied to the leaf sequent in a branch Θ of a pseudo-derivation if the principal formula of the application occurs as the principal formula of a lower application of a destructive rule in Θ.
(2) A non-destructive rule is not applied to the leaf sequent in a branch of a pseudo-derivation if the premise sequent of the application is identical to the conclusion sequent of the application.

It follows that the book-keeping machinery needed to record that a destructive tableau rule has been applied to a formula occurrence with respect to a certain branch is not used in the case with Gentzen rules (but for each application of a destructive Gentzen rule, we do strictly speaking need to record the form of the principal formula).

Given this, all the definitions and results in Section 5 are reformulated. In particular, Definition 5.10 is reformulated to the definition below.

Definition 6.1. *Let b and a be nominals occurring at a branch Θ of a pseudo-derivation. The nominal a is* included *in the nominal b with respect to Θ if for any subformula ϕ of a formula in the end-sequent, if the formula $a : \phi$ occurs in an antecedent (succedent) formula in some sequent in Θ, then $b : \phi$ also occurs in an antecedent (succedent) formula in some sequent in Θ. If a is included in b with respect to Θ, and the lowest sequent with an occurrence of b is lower in the branch than the lowest sequent with an occurrence of a, then we write $a \subseteq_\Theta b$.*

Definition 6.2. *Let $\Gamma \vdash \Delta$ be the sequent whose validity we have to decide. We define by induction a sequence π_0, π_1, π_2, ... of finite pseudo-derivations, each of which is embedded in all its successors. Let π_0 be the finite pseudo-derivation constituted by the single sequent $\Gamma \vdash \Delta$. Assume that the finite pseudo-derivation π_n is defined. If possible, apply an arbitrary rule with the following restrictions:*

- *The existential rule $(\Diamond L)$ is not applied to a leaf sequent $a : \Diamond\phi, \Gamma' \vdash \Delta'$ in a branch Θ if there exists a nominal b such that $a \subseteq_\Theta b$ (and analogously for the existential rule (EL)).*

Let π_{n+1} be the resulting pseudo-derivation.

Compare to the definition of the tableau construction algorithm, Definition 5.11. By reformulating Blackburns tableau system as a Gentzen system, and sketching a decision procedure based on the Gentzen system, we have shown that the loop-check technique does not depend on the particular features of the tableau system, but can be applied in connection with other kinds of proof procedures as well.

7. Related work

In ordinary modal logic, loop-checks are used in connection with standard Fitting-style prefixed tableau systems for transitive logics such as $\mathsf{K4}$, see [7] and [11]. The loop-check technique can be tracked to [6], although a similar idea was involved in a graphical formalism for deciding validity of modal-logical formulas in the earlier book [10]. 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, 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 $\mathsf{K4}$ is obtained, then a systematic tableau construction may not terminate. 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. A recent example is the decision procedure given in [8] which is based on a prefixed tableau system that uses metalinguistic prefixes and accessibility formulas like our Tzakova-style system. 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 and inverse roles together with role hierarchies, which is the case with the logic in [8], then general concept inclusion axioms can be internalised into concepts, as described in [9], 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

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

is defined ensuring that the role U is a transitive and symmetric role containing all the other roles. In the terminology of modal logic, a model satisfying the axioms has the property that the submodel generated by a world w is identical to the equivalence class of w wrt. the equivalence relation obtained by taking the reflexive closure of U. Consequently, a formula $\psi \vee \exists U.\psi$ is true at a world w if and only if ψ is true somewhere at the submodel generated by w, and furthermore, all worlds in the submodel generated by w generates the same submodel. It follows that a formula ϕ is satisfiable wrt. arbitrary models if and only if the formula ϕ' obtained by replacing any universal modality $E\psi$ in ϕ by $\psi \vee \exists U.\psi$ is satisfiable wrt. models satisfying the axioms. In case nominals are involved, further axioms have to be added such that $\psi \vee \exists U.\psi$ is true at a world w if and only if ψ is true somewhere at 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 and inverse roles as well as 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 implicitly 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 and an anonymous referee on the workshop version of the paper for comments. The authors are partially supported by the Danish Natural Science Research Council in connection with the HyLoMOL project.

References

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

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

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

[4] T. Bolander and T. Braüner. Two tableau-based decision procedures for hybrid logic. In H. Schlingloff, editor, *4th workshop "Methods for Modalities" (M4M), Informatik-Bericht Nr. 194*, pages 79–96. Humboldt-Universität zu Berlin, 2005.

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

[6] M. Fitting. *Proof Methods for Modal and Intuitionistic Logic*. Reidel, 1983.

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

[8] I. Horrocks and U. Sattler. A tableaux decision procedure for $\mathcal{SHOIQ}$. In L. P. Kaelbling and A. Saffiotti, editors, *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, pages 448–453, 2005.

[9] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In *Proceedings of LPAR'99*, volume 1705 of *Lecture Notes in Artificial Intelligence*, pages 161–180. Springer-Verlag, 1999.

[10] G.E. Hughes and M.J. Cresswell. *An Introduction to Modal Logic*. Methuen, 1968.

[11] F. Massacci. Single step tableaux for modal logics. *Journal of Automated Reasoning*, 24:319–364, 2000.

[12] J. Seligman. Internalisation: The case of hybrid logics. *Journal of Logic and Computation*, 11:671–689, 2001. Special Issue on Hybrid Logics. C. Areces and P. Blackburn (eds.).

[13] A.S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*, volume 43 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1996.

[14] M. Tzakova. Tableaux calculi for hybrid logics. In N. V. Murray, editor, *Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX 1999)*, volume 1617 of *Lecture Notes in Artificial Intelligence*, pages 278–292. Springer-Verlag, 1999.

[15] Jan van Eijck. Constraint tableaux for hybrid logics. Manuscript, CWI, Amsterdam, 2002.