TERMINATION FOR HYBRID TABLEAUS

THOMAS BOLANDER AND PATRICK BLACKBURN

Abstract. This article extends and improves work on tableau-based decision methods for hybrid logic by Bolander and Braüner [5]. Their paper gives tableau-based decision procedures for basic hybrid logic (with unary modalities) and the basic logic extended with the global modality. All their proof procedures make use of loop-checks to ensure termination.

Here we take a closer look at termination for hybrid tableaus. We cover both types of system used in hybrid logic: prefixed tableaus and internalised tableaus. We first treat prefixed tableaus. We prove a termination result for the basic language (with n-ary operators) that does not involve loop-checks. We then successively add the global modality and n-ary inverse modalities, show why various different types of loop-check are required in these cases, and then re-prove termination. Following this we consider internalised tableaus. At first sight, such systems seem to be more complex. However we define a internalised system which terminates without loop-checks. It is simpler than previously known internalised systems (all of which require loop-checks to terminate) and simpler than our prefix systems (no non-local side conditions on rules are required).

Keywords: Hybrid logic, modal logic, tableau systems, decision procedures, loop-checks.

1. Introduction

The first tableau system for hybrid logic was presented by Tzakova [8]. It is a prefixed tableau calculus, and covers a number of different hybrid logics (including some undecidable ones). However the termination proof given for the case of basic hybrid logic (which is decidable) is flawed: the rules can give rise to non-terminating computations. In Bolander and Braüner [5], tableau-based decision procedures for basic hybrid logic (with unary modalities) and the basic logic extended with the global modality are presented. The calculi are proved to be both terminating and complete, however termination (even for the basic logic) is ensured by using loop-checks. In the present paper we generalise and simplify these results, and refine the proof methods used. In the first part of the paper we discuss prefixed calculi. We introduce a modified Tzakova-style calculus that handles basic hybrid logic with n-ary modalities and show that it provides a complete and terminating calculus for which loop-checks are not needed. We then extend this system to handle the global modality, and n-ary inverse modalities. As we shall see, both additions make the corresponding tableau calculi non-terminating in their pure form. To regain termination we apply different types of loop-check. We motivate the required checks, and re-prove completeness and termination.

In the second part of the paper we turn to internalised systems, as introduced by Blackburn [3], and show that all results obtained for the prefixed calculi translate easily to this setting. Internalised calculi are often regarded as more complex than prefixed systems, and of interest mainly because they are automatically complete (though not necessarily terminating) when enriched with arbitrary pure axioms. However we provide an internalised tableau which generalises the system of Blackburn [3] to cover n-ary modalities, and simplifies it in crucial respects. The resulting calculus is not only simpler than the one provided by Blackburn [3], it is also simpler than its prefixed cousin: termination is guaranteed without loop-checks, and we do not need non-local side conditions on the tableau rules (which is not the case for our prefixed calculus).

Hybrid logic is a relatively new branch of modal logic, but already several tableau systems have been proposed. However there has been little systematic discussion of the available options, no discussion of n-ary modalities and their inverses, and previous termination proofs have tended to be either unnecessarily complicated or flawed. Throughout the paper we have attempted to bring some order to the discussion. We investigate termination (and completeness) of the tableau system progressively: we start with n-ary modalities, and systematically consider the impact

that nominals, satisfaction statements, the global modality, and n-ary inverse modalities have on termination. As we shall see, the addition of a global modality can be handled using a relatively simple loop-check, whereas n-ary inverse modalities require something more sophisticated. We prove our completeness and termination results using the same cluster of concepts: the most important of these is the notion of an urfather.

2. The basics of hybrid logic

We shall in many cases adopt the terminology of [4] and [1]. The hybrid logic we consider is obtained by adding a second sort of propositional symbols, called *nominals*, to ordinary modal logic. We assume that a set Prop of ordinary propositional symbols and a countably infinite set Nom of nominals are given. The sets are taken to be disjoint. The metavariables $p, q, r, \ldots$, and so on, range over ordinary propositional symbols and $a, b, c, \ldots$, and so on, range over nominals. The semantic difference between ordinary propositional symbols and nominals is that nominals are required to be true at *exactly one* world; that is, a nominal "points to a unique world". A nominal can also play the role of an operator, that is, for any nominal a and any formula ϕ, the expression $a\phi$ is a wellformed formula. The formula $a\phi$ is intended to express that the formula ϕ is true at the world pointed to by a. Such a formula is usually called a *satisfaction statement* in hybrid logic, and it is most often written $@_a\phi$ or $a:\phi$ instead of simply $a\phi$. However the simplified notation $a\phi$ will turn out to have some advantages in this paper when we compare prefixed tableaus with internalised tableaus.

We will consider multi-modal languages with modal operators of arbitrary arity. In the following we will assume that we have fixed n modal operators (diamonds) named $\Diamond_0, \Diamond_1, \ldots, \Diamond_{n-1}$, and that for all $i = 0, 1, \ldots, n-1$ the expression $\rho(i)$ denotes the arity of $\Diamond_i$. We also include the inverses of modal operators. A modal operator $\Diamond_i$ has $\rho(i)$ inverses which we denote $\Diamond^-_{i,1}, \Diamond^-_{i,2}, \ldots, \Diamond^-_{i,\rho(i)}$. As a special case a modal operator $\Diamond_j$ of arity 1 has exactly one inverse $\Diamond^-_{j,1}$, as usual. Finally, we have the *global modality* (or *universal modality*) which is a special unary modal operator denoted E. There is no need to include an inverse of the global modality; it is its own inverse.

The language of our hybrid logic will be called L. It is defined by the following grammar:

$$\text{(L)} \qquad \phi \ ::= p \mid a \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \Diamond_i(\phi_1, \ldots, \phi_{\rho(i)}) \mid \Diamond^-_{i,j}(\phi_1, \ldots, \phi_{\rho(i)}) \mid a\phi \mid E\phi$$

where p is an ordinary propositional symbol, a is a nominal, $i \in \{0, \ldots, n-1\}$, and $j \in \{1, \ldots, \rho(i)\}$. In what follows, the metavariables ϕ, ψ, χ, ... range over formulas. As mentioned above, formulas of the form $a\phi$ are called *satisfaction statements*. The dual modal operators $\Box_i$, $\Box^-_{i,j}$ and the propositional connectives not taken as primitive are defined as usual. We now define models.

Definition 2.1. *A* model *for L is a tuple $(W, (R_i)_{i<n}, V)$ where*

1. *W is a non-empty set.*
2. *For all $i = 0, \ldots, n-1$ the set R_i is a relation on W of arity $\rho(i)+1$.*
3. *For each proposition symbol or nominal s, $V(s)$ is a subset of W. If s is a nominal then $V(s)$ is a singleton set.*

The elements of W are called *worlds*, and for all i the relation R_i is called the *accessibility relation* of the modal operator $\Diamond_i$. The relation $\mathcal{M}, w \models \phi$ is defined inductively, where $\mathcal{M} = (W, (R_i)_{i<n}, V)$ is a model, w is an element of W, and ϕ is a formula of our hybrid logic.

$$\begin{array}{rll}
\mathcal{M}, w \models s & \text{iff} & w \in V(s), \text{ where } s \text{ is either a propositional symbol or a nominal} \\
\mathcal{M}, w \models \neg\phi & \text{iff} & \text{not } \mathcal{M}, w \models \phi \\
\mathcal{M}, w \models \phi \wedge \psi & \text{iff} & \mathcal{M}, w \models \phi \text{ and } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \Diamond_i(\phi_1, \ldots, \phi_{\rho(i)}) & \text{iff} & \text{for some } v_1, \ldots, v_{\rho(i)} \in W,\ (w, v_1, \ldots, v_{\rho(i)}) \in R_i \text{ and} \\
 & & \quad \mathcal{M}, v_k \models \phi_k \text{ for all } k = 1, \ldots, \rho(i) \\
\mathcal{M}, w \models \Diamond^-_{i,j}(\phi_1, \ldots, \phi_{\rho(i)}) & \text{iff} & \text{for some } v_1, \ldots, v_{\rho(i)} \in W, (v_1, \ldots, v_j, w, v_{j+1}, \ldots, v_{\rho(i)}) \in R_i \text{ and} \\
 & & \quad \mathcal{M}, v_k \models \phi_k \text{ for all } k = 1, \ldots, \rho(i) \\
\mathcal{M}, w \models a\phi & \text{iff} & \mathcal{M}, v \models \phi, \text{ where } V(a) = \{v\} \\
\mathcal{M}, w \models E\phi & \text{iff} & \text{for some } v \in W,\ \mathcal{M}, v \models \phi
\end{array}$$

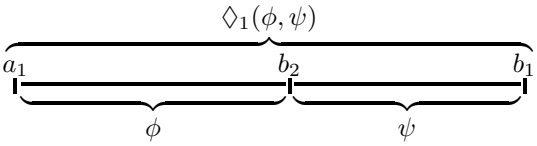


FIGURE 1. Illustration of chop.

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

The semantics given for the inverse modalities $\Diamond^-_{i,j}$ was motivated by the following kind of example. However other options are possible (for example, we could have made use of the *versatile semantics* defined in [9]).

Example 2.2 (Inverse modalities)**.** Interval Temporal Logic (ITL) [7] is a modal logic in which the worlds W are intervals on the real line. For simplicity, we will here let W be the set of proper, closed intervals, that is:

$$W = \{[a, b] \mid a, b \in \mathbb{R} \text{ and } a < b\}.$$

ITL is equipped with a binary modal operator called *chop*, which we will here denote by $\Diamond_1$. The accessibility relation R_1 of this operator is given by

$$R_1 = \{[a_1, b_1], [a_2, b_2], [a_3, b_3] \in W^3 \mid a_2 = a_1 \wedge b_2 = a_3 \wedge b_3 = b_1\}.$$

Given formulas ϕ, ψ and an interval $[a_1, b_1]$ we thus get

$\mathcal{M}, [a_1, b_1] \models \Diamond_1(\phi, \psi)$

$\Leftrightarrow$ for some $[a_2, b_2], [a_3, b_3] \in W$, $([a_1, b_1], [a_2, b_2], [a_3, b_3]) \in R_1$ and $\mathcal{M}, [a_2, b_2] \models \phi$ and $\mathcal{M}, [a_3, b_3] \models \psi$

$\Leftrightarrow$ for some $b_2 \in \mathbb{R}$, $\mathcal{M}, [a_1, b_2] \models \phi$ and $\mathcal{M}, [b_2, b_1] \models \psi$.

Thus, a formula $\Diamond_1(\phi, \psi)$ holds on an interval if and only if this interval can be "chopped" into two subintervals such that ϕ holds on the left of these, and ψ holds on the right. This is illustrated in Figure 1. Now consider the two inverse modalities $\Diamond^-_{1,1}$ and $\Diamond^-_{1,2}$ of $\Diamond_1$. Let T denote any propositional tautology in ITL (e.g. $p \vee \neg p$), let $[a_1, b_1]$ be in W and let ϕ denote any formula. Then we get:

$\mathcal{M}, [a_1, b_1] \models \Diamond^-_{1,1}(T, \phi)$

$\Leftrightarrow$ for some $[a_2, b_2], [a_3, b_3] \in W$, $([a_2, b_2], [a_1, b_1], [a_3, b_3]) \in R_1$ and $\mathcal{M}, [a_2, b_2] \models T$ and $\mathcal{M}, [a_3, b_3] \models \phi$

$\Leftrightarrow$ for some $b_3 > b_1$, $\mathcal{M}, [b_1, b_3] \models \phi$.

Similarly, for $\Diamond^-_{1,2}$ we get:

$\mathcal{M}, [a_1, b_1] \models \Diamond^-_{1,2}(T, \phi)$

$\Leftrightarrow$ for some $[a_2, b_2], [a_3, b_3] \in W$, $([a_2, b_2], [a_3, b_3], [a_1, b_1]) \in R_1$ and $\mathcal{M}, [a_2, b_2] \models T$ and $\mathcal{M}, [a_3, b_3] \models \phi$

$\Leftrightarrow$ for some $a_3 < a_1$, $\mathcal{M}, [a_3, a_1] \models \phi$.

Thus the formula $\Diamond^-_{i,1}(T, \phi)$ holds on an interval if and only if ϕ holds on a right neighbourhood of that interval. Similarly, $\Diamond^-_{i,2}(T, \phi)$ holds on an interval if and only if ϕ holds on a left neighbourhood of that interval. Thus the two standard modalities of Neighbourhood Logic (NL) [10] can in a simple way be encoded in the two inverse modalities of the chop operator.

3. A prefixed tableau calculus

We will now present a prefixed tableau calculus for the hybrid language L. That the tableau calculus is *prefixed* means that the formulas occurring in the tableau rules are *prefixed formulas* on the form $\sigma\phi$, where ϕ is a formula of L and σ belongs to some fixed countably infinite set of symbols called *prefixes*. The set of prefixes will be denoted Pref, and we require that $\mathsf{Nom} \cap \mathsf{Pref} = \emptyset$. The intended interpretation of a prefixed formula $\sigma\phi$ is that σ denotes a world at which ϕ holds.

$$\frac{\sigma\neg a}{\tau a}\,(\neg)^1 \qquad \frac{\sigma\neg\neg\phi}{\sigma\phi}\,(\neg\neg)$$

$$\frac{\sigma(\phi\wedge\psi)}{\begin{array}{c}\sigma\phi\\ \sigma\psi\end{array}}\,(\wedge) \qquad \frac{\sigma\neg(\phi\wedge\psi)}{\sigma\neg\phi \mid \sigma\neg\psi}\,(\neg\wedge)$$

$$\frac{\sigma\Diamond_i(\phi_1,\ldots,\phi_{\rho(i)})}{\begin{array}{c}\sigma\Diamond_i(\sigma_1,\ldots,\sigma_{\rho(i)})\\ \sigma_1\phi_1\\ \vdots\\ \sigma_{\rho(i)}\phi_{\rho(i)}\end{array}}\,(\Diamond)^2 \qquad \frac{\begin{array}{c}\sigma\neg\Diamond_i(\phi_1,\ldots,\phi_{\rho(i)})\\ \sigma\Diamond_i(\sigma_1,\ldots,\sigma_{\rho(i)})\end{array}}{\sigma_1\neg\phi_1 \mid \sigma_2\neg\phi_2 \mid \cdots \mid \sigma_{\rho(i)}\neg\phi_{\rho(i)}}\,(\neg\Diamond)$$

$$\frac{\sigma\Diamond^-_{i,j}(\phi_1,\ldots,\phi_{\rho(i)})}{\begin{array}{c}\sigma_1\Diamond_i(\sigma_2,\ldots,\sigma_j,\sigma,\sigma_{j+1},\ldots,\sigma_{\rho(i)})\\ \sigma_1\phi_1\\ \vdots\\ \sigma_{\rho(i)}\phi_{\rho(i)}\end{array}}\,(\Diamond^-)^2 \qquad \frac{\begin{array}{c}\sigma\neg\Diamond^-_{i,j}(\phi_1,\ldots,\phi_{\rho(i)})\\ \sigma_1\Diamond_i(\sigma_2,\ldots,\sigma_j,\sigma,\sigma_{j+1},\ldots,\sigma_{\rho(i)})\end{array}}{\sigma_1\neg\phi_1 \mid \sigma_2\neg\phi_2 \mid \cdots \mid \sigma_{\rho(i)}\neg\phi_{\rho(i)}}\,(\neg\Diamond^-)$$

$$\frac{\sigma a\phi}{\tau a,\tau\phi}\,(@)^1 \qquad \frac{\sigma\neg a\phi}{\tau a,\tau\neg\phi}\,(\neg@)^1$$

$$\frac{\sigma E\phi}{\tau\phi}\,(E)^1 \qquad \frac{\sigma\neg E\phi}{\gamma\neg\phi}\,(\neg E)^3$$

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

[1] The prefix τ is new to the tableau.
[2] The prefixes $\sigma_1,\ldots,\sigma_{\rho(i)}$ are all new to the tableau.
[3] The prefix γ is already on the branch.

FIGURE 2. Prefixed tableau calculus for the hybrid language L.

In addition to prefixed formulas, the tableau rules contain *accessibility formulas* on the form $\sigma\Diamond_i(\sigma_1,\ldots,\sigma_{\rho(i)})$ where σ and $\sigma_1,\ldots,\sigma_{\rho(i)}$ are prefixes and $i\in\{0,\ldots,n-1\}$. The intended interpretation of $\sigma\Diamond_i(\sigma_1,\ldots,\sigma_{\rho(i)})$ is that the tuple of worlds denoted by $(\sigma_1,\ldots,\sigma_{\rho(i)})$ is accessible from the world denoted by σ by the accessibility relation R_i. In the following we will use the term *formula* to denote either a formula of L, a prefixed formula, or an accessibility formula. The tableau rules of the calculus are given in Figure 2. A *tableau* in this calculus is simply a wellfounded, finitely branching tree in which each node is labeled by a formula, and the edges represent applications of tableau rules in the usual way.

Example 3.1 (A simple tableau)**.** A simple example of a tableau is given in Figure 3. In this tableau there is only one modal operator, so we allow ourselves to drop the index i on $\Diamond$. The modal operator $\Diamond$ is unary. The tableau consists of a single branch.

$\sigma_0(a \wedge \Diamond(a \wedge \neg\Diamond a))$

$|$ ($\wedge$) rule

$\sigma_0 a$

$\sigma_0\Diamond(a \wedge \neg\Diamond a)$

$|$ ($\Diamond$) rule

$\sigma_0\Diamond\sigma_1$

$\sigma_1(a \wedge \neg\Diamond a)$

$|$ ($\wedge$) rule

$\sigma_1 a$

$\sigma_1\neg\Diamond a$

$|$ (Id) rule on $\sigma_1\neg\Diamond a$, $\sigma_0 a$, $\sigma_1 a$

$\sigma_0\neg\Diamond a$

$|$ ($\neg\Diamond$) rule on $\sigma_0\neg\Diamond a$, $\sigma_0\Diamond\sigma_1$

$\sigma_1\neg a$

FIGURE 3. A simple tableau.

The rules ($\neg$), ($\Diamond$), ($\Diamond^-$), (@), ($\neg$@), and (E) are called *prefix generating rules*. Whenever one of these rules is applied on a branch, at least one new prefix will be introduced to the branch. We impose two general constraints on the construction of tableaus:

- A prefix generating rule is never applied twice to the same premise on the same branch.
- A formula is never added to a tableau branch where it already occurs.

A *saturated tableau* is a tableau in which no more rules can be applied that satisfy the constraints. A *saturated branch* is a branch of a saturated tableau. A branch of a tableau is called *closed* if it contains formulas $\sigma\phi$ and $\sigma\neg\phi$ for some σ and ϕ. Otherwise the branch is called *open*. A *closed tableau* is one in which all branches are closed, and an *open tableau* is one in which at least one branch is open. Below we will consider several subsystems of the calculus of Figure 2 where only a subset of the rules are allowed to be applied. In such subsystems, a *saturated tableau* is of course simply a tableau in which none of the rules in the subset can be applied.

Definition 3.2. *When a prefixed formula $\sigma\phi$ occurs in a tableau branch Θ we will write $\sigma\phi \in \Theta$, and say that ϕ is* true *at σ on Θ or that σ makes ϕ* true *on Θ.*

Given a tableau branch Θ and a prefix σ the *set of true formulas* at σ on Θ, written $T^\Theta(\sigma)$, then becomes

$$T^\Theta(\sigma) = \{\phi \mid \sigma\phi \in \Theta\}.$$

In the following when we say *tableau* we will always mean a tableau constructed from (some subset of) the rules of Figure 2. A formula ϕ is said to be a *quasi-subformula* of a formula ψ if one of the following holds:

- ϕ is a subformula of ψ.
- ϕ has the form $\neg\chi$, where χ is a subformula of ψ.

Lemma 3.3 (**Quasi-subformula Property**). *Let $\mathcal{T}$ be a tableau with the prefixed formula $\sigma_0\phi_0$ as root. For any prefixed formula $\sigma\phi$ occurring on $\mathcal{T}$, ϕ is a quasi-subformula of ϕ_0.*

Proof. This is easily seen by going through each of the tableau rules of Figure 2. □

Lemma 3.4. *Let Θ be a branch of a tableau, and let σ be any prefix occurring on Θ. The set $T^\Theta(\sigma)$ is finite.*

Proof. Let $\sigma_0\phi_0$ denote the first formula on Θ, that is, the root of the tableau. From the Quasi-subformula Property, Lemma 3.3, we get that

$$T^{\Theta}(\sigma) \subseteq \{\phi \mid \phi \text{ is a subformula of } \phi_0\} \cup \{\neg\phi \mid \phi \text{ is a subformula of } \phi_0\}.$$

Since ϕ_0 has only got finitely many subformulas this proves $T^{\Theta}(\sigma)$ to be finite. □

When given a tableau calculus for a language with a given semantics, one usually wants to prove *soundness* and *completeness* of the calculus with respect to the semantics. Furthermore, if the tableau calculus is supposed to give a decision procedure for the logic, one will have to prove *termination* of the calculus, that is, show that there is a terminating algorithm for applying the tableau rules that preserves completeness. In some cases, more detailed analysis of the algorithm may give rise to information about the *complexity* of the logic.

In the following, we will prove the three properties *soundness*, *completeness*, and *termination* for the prefixed tableau calculus of L. We will do this in steps. First we prove that the three properties hold for a simple fragment of the calculus, and then we step by step add more tableau rules and at each step show that the properties still hold. As we add more and more tableau rules we will need more and more complex tableau construction algorithms in order to ensure termination—but the tableau construction algorithm used at each step will always be a rather simple extension of the algorithm used at the previous step. For the simpler modal and hybrid languages we consider, the tableau algorithms we define are not optimal from a complexity theoretic perspective. In our approach, open tableau branches are complete descriptions of satisfying models, hence branches may be exponential in the length of the input formula. However basic modal logic and basic hybrid logic are both known to be PSPACE-complete (see [4]) so more space-efficient algorithms exist. But while it would be of some interest to define such algorithms, the tableau systems defined below have the advantage that they extend relatively straightforwardly to systems for richer languages containing the universal modality and inverse modalities. For such logics, exponential branches are unavoidable, as these logics are known to be EXPTIME-complete. For more on the complexity of hybrid logic, see [1] and [2].

4. Termination without loop-checks

4.1. **Propositional logic.** We start out with the simplest thing imaginable: a tableau system for ordinary propositional logic. This is obtained by restricting the calculus for L to the language L_1 given by the following grammar:

$$(L_1) \qquad \phi \ ::= \ p \mid \neg\phi \mid \phi_1 \wedge \phi_2$$

The calculus for this language only consists of the rules $(\neg\neg)$, $(\wedge)$, and $(\neg\wedge)$ of Figure 2.

$$(L_1 \text{ rules}) \qquad (\neg\neg), (\wedge), (\neg\wedge)$$

Soundness and completeness for this fragment are simple and well-known results. Termination is almost immediate: whenever a rule is applied, the formula lengths of the conclusions are strictly smaller than the formula length of the premise. Thus when we move down through a branch of a tableau the formula lengths will be strictly decreasing. Therefore a tableau branch cannot be of infinite length, and thus a tableau cannot be infinite either, since it is only finitely branching.

4.2. **Adding modal operators.** The first step from L_1 towards hybrid logic is to add modal operators, that is, to consider the language L_2 given by the following grammar:

$$(L_2) \qquad \phi \ ::= \ p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \Diamond_i(\phi_1, \ldots, \phi_{\rho(i)})$$

The tableau calculus for this language consists of the rules for L_1 extended with the rules $(\Diamond)$ and $(\neg\Diamond)$.

$$(L_2 \text{ rules}) \qquad (\neg\neg), (\wedge), (\neg\wedge), (\Diamond), (\neg\Diamond)$$

Soundness and completeness with respect to the given semantics are still simple and well-known properties, but now termination requires a little more work. The conclusion of rules still all have strictly smaller length than the premises, but the rule $(\neg\Diamond)$ can applied to the same prefixed formula $\sigma\neg\Diamond_i(\phi_1, \ldots, \phi_{\rho(i)})$ on the same branch many times during the course of constructing a

tableau, so we can not be sure that formula lengths will be strictly monotonically decreasing when moving down through the branch. Therefore we need a slightly more elaborate argument in order to prove termination. However, the fundamental underlying idea is still to ensure termination by showing that the length of formulas will be decreasing through the course of constructing a tableau. This is actually the underlying idea in all of our termination proofs presented in this article.

Definition 4.1. *Let Θ be a branch of a tableau. If a prefix τ has been introduced to the branch by applying one of the prefix generating rules to a premise $\sigma\phi$ then we say that τ is* generated *by σ on Θ, and we write $\sigma \prec_\Theta \tau$. We use $\prec_\Theta^*$ to denote the transitive and reflexive closure of the relation $\prec_\Theta$.*

Lemma 4.2. *Let Θ be a branch of a tableau. The graph $G = (N^\Theta, \prec_\Theta)$, where N^Θ is the set of prefixes occurring on Θ, is a wellfounded, finitely branching tree.*

Proof. That G is wellfounded follows from the observation that if $\sigma \prec_\Theta \tau$, 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 G is finitely branching follows from the fact that for any given prefix σ the set $T^\Theta(\sigma)$ is finite (Lemma 3.4), and for each of formula $\phi \in T^\Theta(\sigma)$ at most $\max\{\rho(i) \mid 0 \leq i < n\}$ new prefixes can have been generated from σ (by applying one of the prefix generating rules to $\sigma\phi$). Thus G is a wellfounded, finitely branching tree. □

Lemma 4.3. *Let Θ be a branch of a tableau. Then Θ is infinite if and only if there exists an infinite chain of prefixes*

$$\sigma_1 \prec_\Theta \sigma_2 \prec_\Theta \sigma_3 \prec_\Theta \cdots .$$

Proof. The 'if' direction is trivial. To prove the 'only if' direction, let Θ be any infinite tableau branch. Let $G = (N^\Theta, \prec_\Theta)$ be defined as in Lemma 4.2 above. According to the lemma, G is a wellfounded, finitely branching tree. We will furthermore prove that G is infinite. Note that according to our tableau conventions all prefixed formulas occurring on the infinite branch Θ are distinct. Since for each prefix σ there can only be finitely many distinct formulas $\sigma\phi$ occurring on Θ (Lemma 3.4), this implies that infinitely many distinct prefixes occur on Θ. Thus G must be infinite. Since we now know that G is an infinite, wellfounded, finitely branching tree, we also know that it must contain an infinite path (König's Lemma), that is, a path $\sigma_1 \prec_\Theta \sigma_2 \prec_\Theta \sigma_3 \prec_\Theta \cdots$. □

The above lemma will be applied a number of times in the following. First we will apply it to prove termination of the tableau calculus of L_2.

Definition 4.4. *Let Θ be a branch of a tableau, and let σ be a prefix occurring on Θ. We define $m_\Theta(\sigma)$ by*

$$m_\Theta(\sigma) = \max\{|\phi| \mid \sigma\phi \in \Theta\},$$

where $|\phi|$ is the length of the formula ϕ.

Thus for all branches Θ and all prefixes σ occurring on Θ, the number $m_\Theta(\sigma)$ is the maximal length of formulas true at σ on Θ.

Lemma 4.5 (Decreasing length). *Let Θ be a branch of a tableau, and let σ and τ be prefixes occurring on Θ such that each formula true at τ has been introduced by applying one of the rules of L_2: $(\neg\neg)$, $(\wedge)$, $(\neg\wedge)$, $(\Diamond)$, or $(\neg\Diamond)$. If $\sigma \prec_\Theta \tau$ then $m_\Theta(\sigma) > m_\Theta(\tau)$.*

Proof. Assume $\sigma \prec_\Theta \tau$. Let ϕ be a formula of maximal length true at τ on Θ. We need to prove $m_\Theta(\sigma) > |\phi|$. By assumption, the prefixed formula $\tau\phi$ must have been introduced on Θ by applying one of the rules $(\neg\neg)$, $(\wedge)$, $(\neg\wedge)$, $(\Diamond)$, or $(\neg\Diamond)$. It can however not have been introduced by applying any of the rules $(\neg\neg)$, $(\wedge)$ or $(\neg\wedge)$, since this contradicts the maximality of ϕ. Thus $\tau\phi$ must have been introduced by an application of either $(\Diamond)$ or $(\neg\Diamond)$. In the case of $(\Diamond)$, $\tau\phi$ must be introduced by applying the $(\Diamond)$ rule to a premise of the form $\sigma\Diamond_i(\ldots, \phi, \ldots)$ since τ is

generated by σ. In the case of $(\neg\Diamond)$, $\tau\phi$ must be on the form $\tau\neg\psi$ and introduced by applying the $(\neg\Diamond)$ rule to a pair of premises on the form

$$\sigma'\neg\Diamond_i(\dots,\psi,\dots),\ \sigma'\Diamond_i(\dots,\tau,\dots).$$

Since $\sigma'\Diamond_i(\dots,\tau,\dots)$ occurs on Θ, τ must be generated by σ' (the rule $(\Diamond)$ is the only one producing accessibility formulas). Therefore $\sigma'=\sigma$. In both cases we see that $\tau\phi$ is introduced by applying a rule to a formula $\sigma\chi$ where χ has greater length than ϕ. Thus we get

$$m_\Theta(\sigma)\geq|\chi|>|\phi|\,,$$

as needed. □

Termination of the tableau calculus of L_2 now follows immediately from Lemma 4.3 and 4.5, as shown below.

Theorem 4.6 (**Termination of L_2**). *Any tableau in the calculus of L_2 is finite.*

Proof. Assume there exists an infinite tableau of L_2. Then it must have an infinite branch Θ. By Lemma 4.3, there exists an infinite chain

$$\sigma_1\prec_\Theta\sigma_2\prec_\Theta\sigma_3\prec_\Theta\cdots.$$

Now by Lemma 4.5 we have

$$m_\Theta(\sigma_1)>m_\Theta(\sigma_2)>m_\Theta(\sigma_3)>\cdots$$

which is a contradiction, since $m_\Theta(\sigma)$ is a non-negative number for any prefix σ. □

Since we already know the calculus of L_2 to be sound and complete, the theorem shows that the calculus gives a decision procedure for the logic. Note how the proof above is related to the termination proof of the calculus for L_1. The idea is still to show that formula lengths are strictly decreasing down through a path (and that the path therefore cannot be infinite), but now the path is not the tableau branch Θ itself but rather a path in the graph $G=(N^\Theta,\prec_\Theta)$. Thus the basic idea underlying the termination proof in the case of L_2 is the same as in the simple case of L_1.

4.3. **Adding nominals.** Now consider extending L_2 to include the most basic element of hybrid logics: the nominals. This gives us a language L_3 defined by the following grammar:

$$(L_3)\qquad \phi\ ::=\ p\mid a\mid\neg\phi\mid\phi_1\wedge\phi_2\mid\Diamond_i(\phi_1,\dots,\phi_{\rho(i)})$$

Syntactically this extension of L_2 simply amounts to introducing a second sort of propositional symbols which we have chosen to call nominals. The tableau calculus of L_2—that is, the rules $(\neg\neg)$, $(\wedge)$, $(\neg\wedge)$, $(\Diamond)$, and $(\neg\Diamond)$—does *not* give a complete proof theory for L_3. The reason is simply that according to the semantics the nominals should be treated in a special way—they should be true at one and only one world—but we have not yet added any rules that will treat the nominals differently from all the other propositional symbols. To deal with the special treatment of the nominals we add the rules $(\neg)$ and (Id) of Figure 2 to form a calculus for L_3. It is easy to see that this extension is sound with respect to the semantics. It is also possible to show that the calculus is complete, but we will not do that here, since unfortunately the calculus turns out not to be terminating. The reason is that when we extend with the rule (Id) then Lemma 4.5 no longer holds. This is shown by the following simple example.

Example 4.7 (Non-termination). Consider the hybrid formula $a\wedge\Diamond a$, where a is a nominal and $\Diamond$ is a unary modal operator. The formula belongs to the language L_3. It is possible to make an infinite tableau branch Θ in the calculus of L_3 with this formula as root. This is shown in Figure 4. From the figure we see that

$$\sigma_0\prec_\Theta\sigma_1\prec_\Theta\sigma_2\prec_\Theta\cdots.$$

However, at the same time we have

$$|\Diamond a|=m_\Theta(\sigma_1)=m_\Theta(\sigma_2)=\cdots.$$

Thus the infinite branch Θ is a counter-example to Lemma 4.5 holding when the (Id) rule is

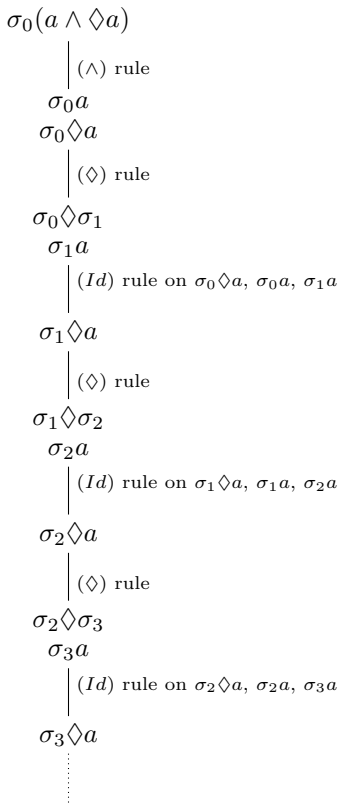


FIGURE 4. An infinite tableau using the (Id) rule.

added to the calculus.

The example shows that we do not have termination when we add the rule (Id). Let us try to analyse the problem a bit further. First a new definition.

Definition 4.8. *Let Θ be a branch of a tableau. Define a binary relation $\sim_\Theta$ on the prefixes occurring on Θ by*

$$\sigma \sim_\Theta \tau \quad \textit{iff} \quad \textit{there exists a nominal } a \textit{ such that both } \sigma a \textit{ and } \tau a \textit{ occur on } \Theta.$$

In other words, we define $\sigma \sim_\Theta \tau$ to hold whenever there exists a nominal that both σ and τ make true on Θ. The reflexive closure of $\sim_\Theta$ will be denoted $\sim_\Theta^=$. Thus $\sigma \sim_\Theta^= \tau$ holds if either $\sigma = \tau$ or there is a nominal that they both make true.

Let Θ be a branch of a tableau, and assume $\sigma \sim_\Theta \tau$. By definition, $\sigma \sim_\Theta \tau$ means that there is a nominal a that σ and τ both make true on Θ. This implies that σ and τ must denote the same world in the intended model, since according to the semantics, nominals are true at a unique world. If some formula $\sigma\phi$ occurs on Θ, then by the (Id) rule we can extend the branch with $\tau\phi$. In other words, any formula true at σ on Θ can be made true at τ by a suitable extension of the branch. This shows what the problem with the (Id) rule is. It is a rule that allows us to "copy information between worlds": if σ and τ are prefixes with $\sigma \sim_\Theta \tau$ then any formula true at σ can be copied to τ. This breaks down the argument of decreasing lengths of formulas used to prove termination of the calculus of L_2: we can not make sure that the maximal lengths of formulas will be strictly decreasing down through a chain $\sigma_1 \prec_\Theta \sigma_2 \prec_\Theta \sigma_3 \prec_\Theta \cdots$ since if for instance $\sigma_1 \sim_\Theta \sigma_3$ then any formula true at σ_1 can be copied to σ_3.

To obtain a terminating proof procedure for L_3 we will try to restrict the rule (Id) in a way that will allow the decreasing length argument to go through. The general idea is that if σ_i and

σ_{i+j} are $\sim_\Theta$-related prefixes on a chain

$$\sigma_1 \prec_\Theta \sigma_2 \prec_\Theta \cdots \prec_\Theta \sigma_i \prec_\Theta \cdots \prec_\Theta \sigma_{i+j} \prec_\Theta \cdots$$

then we should allow formulas to be copied from σ_{i+j} to σ_i but not from σ_i to σ_{i+j}. In other words, copying with the (Id) rule should respect the ordering of the prefixes by the relation $\prec_\Theta$. We define the new restricted (Id) rule, (νId), by:

$$\boxed{\frac{\sigma\phi, \sigma a, \tau a}{\tau\phi}\,(\nu Id) \quad \begin{array}{l}\tau \text{ is the earliest introduced} \\ \text{prefix making } a \text{ true}\end{array}}$$

In addition to this rule, we need to allow nominals to be copied arbitrarily between equivalent prefixes. This is taken care of by adding the following rule:

$$\boxed{\frac{\sigma b, \sigma a, \tau a}{\tau b}\,(Nom)}$$

Note that both (νId) and (Nom) are simply restrictions of the (Id) rule. If the rule (Id) is replaced by (νId) and (Nom) it is easy to see that we can no longer make an infinite tableau with root $\sigma(a \wedge \Diamond a)$ as we did in Example 4.7. We will prove termination of the calculus of L_3 with the (νId) and (Nom) rules. In the following we will use the expression *the calculus of* L_3 to refer to the calculus consisting of the following rules:

(L_3 rules) $\qquad (\neg), (\neg\neg), (\wedge), (\neg\wedge), (\Diamond), (\neg\Diamond), (\nu Id), (Nom)$

First we prove a strengthening of Lemma 4.5.

Lemma 4.9 (Decreasing length). *Let Θ be a branch of a tableau, and let σ and τ be prefixes occurring on Θ such that each formula true at τ has been introduced by applying one of the rules $(\neg)$, $(\neg\neg)$, $(\wedge)$, $(\neg\wedge)$, $(\Diamond)$, $(\neg\Diamond)$, or (Nom). If $\sigma \prec_\Theta \tau$ then $m_\Theta(\sigma) > m_\Theta(\tau)$.*

Proof. Assume $\sigma \prec_\Theta \tau$. Let ϕ be a formula of maximal length true at τ on Θ. We need to prove $m_\Theta(\sigma) > |\phi|$. The cases where $\tau\phi$ has been introduced by an application of a rule other than $(\neg)$ and (Nom) have already been dealt with in the proof of Lemma 4.5. Thus assume that $\tau\phi$ has been introduced by an application of either $(\neg)$ or (Nom). In both cases ϕ must have length 1. Since the prefix σ has generated the prefix τ, σ must make at least one formula of length > 1 true (no prefix generating rules apply to formulas of length 1). Thus we have $m_\Theta(\sigma) > 1 = |\phi|$, as needed. □

Theorem 4.10 (Termination of the calculus of L_3). *Any tableau in the calculus of L_3 is finite.*

Proof. Assume to obtain a contradiction that there exists an infinite tableau $\mathcal{T}$ in the calculus of L_3. Let Θ be an infinite branch of $\mathcal{T}$. Then, by Lemma 4.3, there must exist an infinite chain of prefixes

$$\sigma_1 \prec_\Theta \sigma_2 \prec_\Theta \sigma_3 \prec_\Theta \cdots . \tag{1}$$

Note that $\mathcal{T}$ contains only finitely many nominals: none of the considered tableau rules are generating new nominals, so the number of nominals on $\mathcal{T}$ must be the number of nominals occurring in the root formula. For each nominal a on $\mathcal{T}$ let τ_a denote the earliest introduced prefix on $\mathcal{T}$ making a true. Whenever the rule (νId) is applied on Θ it produces a conclusion of the form $\tau_a\phi$ for some nominal a. Since the number of nominals is finite, the number of such prefixes τ_a must also be finite. Thus there exists an infinite subchain

$$\sigma_i \prec_\Theta \sigma_{i+1} \prec_\Theta \sigma_{i+2} \prec_\Theta \cdots$$

of (1) containing none of the prefixes of the form τ_a. Thus none of the formulas true at $\sigma_i, \sigma_{i+1}, \ldots$ have been introduced using the (νId) rule. We can therefore apply Lemma 4.9 to conclude that

$$m_\Theta(\sigma_i) > m_\Theta(\sigma_{i+1}) > m_\Theta(\sigma_{i+2}) > \cdots .$$

This is a contradiction. □

We will now prove that the calculus of L_3 is also complete. To do this we need a few new notions.

Definition 4.11. *Let Θ be a branch of a tableau, and let σ be a prefix occurring on Θ. The* nominal urfather *of σ on Θ, written $s_\Theta(\sigma)$, is defined to be the earliest introduced prefix τ on Θ for which $\tau \sim_\Theta^= \sigma$. In other words, $s_\Theta(\sigma)$ is defined by*

1. *If no nominals are true at σ on Θ, then $s_\Theta(\sigma) = \sigma$.*
2. *Otherwise $s_\Theta(\sigma)$ is the earliest introduced prefix on Θ which makes some nominal true that σ also makes true.*

A prefix σ is called a nominal urfather *on Θ if $\sigma = s_\Theta(\tau)$ for some prefix τ.*

Lemma 4.12 (Urfather Closure). *Let Θ be a saturated branch in a calculus containing at least (νId). If $\sigma\phi$ occurs on Θ then $s_\Theta(\sigma)\phi$ also occurs on Θ.*

Proof. Assume $\sigma\phi \in \Theta$. If $s_\Theta(\sigma) = \sigma$ then there is nothing to prove. So assume $s_\Theta(\sigma) \neq \sigma$. In that case the definition of $s_\Theta(\sigma)$ gives us the existence of a nominal a true at both σ and $s_\Theta(\sigma)$, where $s_\Theta(\sigma)$ is the earliest introduced prefix making a true. Since Θ is saturated we have closure under the (νId) rule. Since Θ contains all of $\sigma\phi$, σa and $s_\Theta(\sigma)a$, closure under the (νId) gives us $s_\Theta(\sigma)\phi \in \Theta$. □

Lemma 4.13. *Let Θ be a saturated branch in a calculus containing at least (Nom), and let σ and τ be nominals occurring on Θ. $\sigma \sim_\Theta \tau$ if and only if σ and τ make the same non-empty set of nominals true on Θ.*

Proof. The 'if' direction follows immediately from the definition of $\sim_\Theta$. We thus turn to the 'only if' direction. If $\sigma \sim_\Theta \tau$ then there is a nominal a that both σ and τ make true on Θ. Let b be any nominal true at σ. Since Θ is closed under the (Nom) rule and it contains all of σb, σa and τa it must also contain τb. This proves that every nominal true at σ is also true at τ. The other direction is by symmetry. □

Lemma 4.14 (Urfather Equality). *Let Θ be a saturated branch in a calculus containing at least (Nom). If $\sigma \sim_\Theta \tau$ then $s_\Theta(\sigma) = s_\Theta(\tau)$.*

Proof. Assume $\sigma \sim_\Theta \tau$. Then there is a nominal a such that $\sigma a, \tau a \in \Theta$. By definition, $s_\Theta(\sigma)$ is the earliest introduced prefix that makes some nominal true which σ also makes true. Correspondingly, $s_\Theta(\tau)$ is the earliest introduced prefix making some nominal true which τ also makes true. Since by Lemma 4.13, σ and τ make the same set of nominals true, the two prefixes $s_\Theta(\sigma)$ and $s_\Theta(\tau)$ must be identical. □

The two last lemmata above are important. The former, Lemma 4.13, shows that $\sim_\Theta^=$ must be an equivalence relation whenever Θ is closed under all applications of the (Nom) rule (Lemma 4.13 shows that the relation $\sim_\Theta$ must be symmetric and transitive, and taking the reflexive closure of this relation we then get an equivalence relation). The latter, Lemma 4.14, shows that for each equivalence class under this relation there is a unique nominal urfather. This urfather is of course itself a member of the equivalence class. So urfathers are a kind of 'privileged members' of the equivalence classes under $\sim_\Theta^=$: for each equivalence class we can choose the urfather of that class as a representative. This will allow us, as we will see later, to construct a model from an open saturated branch using the set of urfathers as the set of worlds of that model. As we noted above, when two prefixes are $\sim_\Theta$-related they denote identical worlds in the intended model, so it is important that we only make one world out of each equivalence class—the 'urfather world'. Alternatively, one could make a model out of the equivalence classes themselves, but that will not make things any simpler.

Let Θ be a tableau branch with root $\sigma_0\phi_0$. Note that then we have $s_\Theta(\sigma_0) = \sigma_0$, implying that σ_0 is a nominal urfather on Θ. Thus the root prefix of a tableau branch is always a nominal urfather on that branch. More generally, any prefix σ for which $s_\Theta(\sigma) = \sigma$ will be a nominal urfather on Θ. It also holds the other way around, as the following lemma shows:

Lemma 4.15 (Urfather Characterisation). *Let Θ be a saturated branch in a calculus containing at least (Nom). Then σ is a nominal urfather on Θ if and only if $s_\Theta(\sigma) = \sigma$.*

Proof. The 'if' direction immediately follows from the definition of a nominal urfather. So let us consider the 'only if' direction. If σ is a nominal urfather then $s_\Theta(\tau) = \sigma$ for some prefix τ. If $\tau = \sigma$ then the proof is complete. Otherwise $\sigma \sim_\Theta \tau$, by definition of s_Θ. Urfather Equality, Lemma 4.14, then implies $s_\Theta(\sigma) = s_\Theta(\tau)$. Since we also have $s_\Theta(\tau) = \sigma$, we thus get $s_\Theta(\sigma) = \sigma$, as required. □

Given an open, saturated branch Θ with root $\sigma_0\phi_0$, we now define a model $\mathcal{M}^\Theta$ by

$$\begin{aligned}
\mathcal{M}^\Theta &= (W^\Theta, (R_i^\Theta)_{i<n}, V^\Theta), \text{ where} \\
W^\Theta &= \{\sigma \mid \sigma \text{ is a prefix occurring on } \Theta\} \\
R_i^\Theta &= \{(\sigma, s_\Theta(\sigma_1), \ldots, s_\Theta(\sigma_{\rho(i)})) \mid \sigma\Diamond_i(\sigma_1, \ldots, \sigma_{\rho(i)}) \text{ occurs on } \Theta\} \\
V^\Theta(p) &= \{\sigma \mid \sigma p \text{ occurs on } \Theta\} \\
V^\Theta(a) &= \begin{cases} \{\sigma_0\} & \text{if there is no } \sigma \text{ for which } \sigma a \in \Theta. \\ \{s_\Theta(\sigma)\} & \text{if } \sigma a \in \Theta. \end{cases}
\end{aligned}$$

In this definition, p is any ordinary propositional symbol and a is any nominal. That $V^\Theta(a)$ is uniquely defined for any nominal a follows directly from Urfather Equality (Lemma 4.14). We are now ready for the completeness proof.

Theorem 4.16 (Completeness of the calculus of L_3). *Let Θ be an open, saturated branch in the calculus of L_3. For any formula $\sigma\phi$ on Θ where σ is a nominal urfather we have $\mathcal{M}^\Theta, \sigma \models \phi$. In particular, we have $\mathcal{M}^\Theta, \sigma_0 \models \phi_0$ where $\sigma_0\phi_0$ is the root of the branch.*

Proof. The proof is by induction on the syntactic structure of ϕ. The base cases are $\phi = p$ and $\phi = \neg p$ for propositional symbols p and $\phi = a$ and $\phi = \neg a$ for nominals a. Assume first $\phi = p$ and assume $\sigma p \in \Theta$ where σ is a nominal urfather. Then $\sigma \in V^\Theta(p)$, by definition. This immediately implies $\mathcal{M}^\Theta, \sigma \models p$. Assume now $\phi = \neg p$ and assume $\sigma\neg p \in \Theta$ where σ is a nominal urfather. Then we cannot have $\sigma p \in \Theta$ since Θ is an open branch. Thus we have $\sigma \notin V^\Theta(p)$ which implies $\mathcal{M}^\Theta, \sigma \models \neg p$. Assume now $\phi = a$ and assume $\sigma a \in \Theta$ where σ is a nominal urfather. Then we get $V^\Theta(a) = \{s_\Theta(\sigma)\} = \{\sigma\}$, using Urfather Characterisation (Lemma 4.15). From this it immediately follows that $\mathcal{M}^\Theta, \sigma \models a$. Assume finally $\phi = \neg a$ and assume $\sigma\neg a \in \Theta$ where σ is a nominal urfather. Since Θ is saturated it must also contain a formula τa, by closure under the $(\neg)$ rule. Thus we have $V^\Theta(a) = \{s_\Theta(\tau)\}$. By Urfather Closure (Lemma 4.12) we have $s_\Theta(\tau)a \in \Theta$. Since Θ is thus an open branch containing both $\sigma\neg a$ and $s_\Theta(\tau)a$ we get $\sigma \neq s_\Theta(\tau)$. Thus we have $\sigma \notin V^\Theta(a)$ which implies $\mathcal{M}, \sigma \models \neg a$. This concludes the base cases. We now turn to the induction step. The cases where ϕ is on the form $\neg\neg\psi$, $\psi \wedge \chi$ or $\neg(\psi \wedge \chi)$ are trivial. Consider the case where ϕ is of the form $\Diamond_i(\phi_1, \ldots, \phi_{\rho(i)})$. Assume $\sigma\Diamond_i(\phi_1, \ldots, \phi_{\rho(i)})$ occurs on Θ where σ is a nominal urfather. Then since Θ is closed under applications of the $(\Diamond)$ rule, Θ must also contain formulas of the form:

$$\begin{aligned}
&\sigma\Diamond_i(\sigma_1, \ldots, \sigma_{\rho(i)}) \\
&\sigma_1\phi_1, \sigma_2\phi_2, \ldots, \sigma_{\rho(i)}\phi_{\rho(i)}.
\end{aligned}$$

By Urfather Closure (Lemma 4.12) this implies that Θ contains all of the following formulas as well:

$$s_\Theta(\sigma_1)\phi_1, \ldots, s_\Theta(\sigma_{\rho(i)})\phi_{\rho(i)}.$$

The induction hypothesis now gives us

$$\mathcal{M}^\Theta, s_\Theta(\sigma_1) \models \phi_1, \ldots, \mathcal{M}^\Theta, s_\Theta(\sigma_{\rho(i)}) \models \phi_{\rho(i)}. \tag{2}$$

Since $\sigma\Diamond_i(\sigma_1, \ldots, \sigma_{\rho(i)})$ occurs on Θ, we furthermore get

$$(\sigma, s_\Theta(\sigma_1), s_\Theta(\sigma_2), \ldots, s_\Theta(\sigma_{\rho(i)})) \in R_i^\Theta. \tag{3}$$

Combining (2) and (3) immediately gives us $\mathcal{M}^\Theta, \sigma \models \Diamond_i(\phi_1, \ldots, \phi_{\rho(i)})$, as required. Consider, finally, the case where ϕ is on the form $\neg\Diamond_i(\phi_1, \ldots, \phi_{\rho(i)})$. Assume $\sigma\neg\Diamond_i(\phi_1, \ldots, \phi_{\rho(i)})$ occurs on Θ where σ is a nominal urfather. We need to prove $\mathcal{M}^\Theta, \sigma \models \neg\Diamond_i(\phi_1, \ldots, \phi_{\rho(i)})$. If there are no worlds $\sigma_1, \ldots, \sigma_{\rho(i)}$ such that $(\sigma, \sigma_1, \ldots, \sigma_{\rho(i)}) \in R_i^\Theta$ then this holds trivially. Otherwise, let such $\sigma_1, \ldots, \sigma_{\rho(i)}$ be chosen arbitrarily. By definition of R_i^Θ there must be prefixes $\tau_1, \ldots, \tau_{\rho(i)}$ such that $\sigma_j = s_\Theta(\tau_j)$ for all $j = 1, \ldots, \rho(i)$ and such that Θ contains $\sigma\Diamond_i(\tau_1, \ldots, \tau_{\rho(i)})$. Since Θ is saturated it must contain $\tau_j\neg\phi_j$ for some $j \in \{1, \ldots, \rho(i)\}$ (closure under the $(\neg\Diamond)$ rule). Urfather Closure now gives $\sigma_j\neg\phi_j \in \Theta$ which by induction hypothesis implies $\mathcal{M}^\Theta, \sigma_j \models \neg\phi_j$. From this it follows that $\mathcal{M}^\Theta, \sigma \models \neg\Diamond_i(\phi_1, \ldots, \phi_{\rho(i)})$. □

The conclusion is that when we replace the rule (Id) by (νId) and (Nom) we get a calculus for the language L_3 which is both sound, complete and terminating.

4.4. **Adding satisfaction statements.** Now consider adding satisfaction statements to the language L_3, that is, define a language L_4 by the following grammar:

$$(L_4) \qquad \phi ::= p \mid a \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \Diamond_i(\phi_1, \ldots, \phi_{\rho(i)}) \mid a\phi$$

We will define the *calculus of* L_4 to consist of the rules of the calculus of L_3 extended with (@) and $(\neg@)$.

$$(L_4 \text{ rules}) \qquad (\neg), (\neg\neg), (\wedge), (\neg\wedge), (\Diamond), (\neg\Diamond), (\nu Id), (Nom), (@), (\neg@)$$

We will prove soundness, completeness, and termination of this calculus. Soundness is again simple to prove. Termination is also simple, given that we already now the calculus of L_3 to be terminating. To prove termination we first need a strengthening of Lemma 4.9.

Lemma 4.17 (**Decreasing length**). *Let Θ be a branch of a tableau in the calculus of L_4 not containing any applications of the (νId) rule. If $\sigma \prec_\Theta \tau$ then $m_\Theta(\sigma) > m_\Theta(\tau)$.*

Proof. Assume $\sigma \prec_\Theta \tau$. Let ϕ be a formula of maximal syntactic complexity true at τ on Θ. We need to prove $m_\Theta(\sigma) > |\phi|$. The cases where $\tau\phi$ has been introduced by an application of a rule other than (@) and $(\neg@)$ have already been dealt with in the proof of Lemma 4.9. Thus assume that $\tau\phi$ has been introduced by an application of (@). Then $\tau\phi$ must have been introduced together with a formula τa by applying (@) to a premise of the form $\sigma a\phi$. Thus we get $m_\Theta(\sigma) \geq |a\phi| > |\phi|$, as needed. The case where the rule applied is $(\neg@)$ is treated exactly the same way. □

Theorem 4.18 (**Termination of the calculus of L_4**). *Any tableau in the calculus of L_4 is finite.*

Proof. The proof is the same as the proof of Theorem 4.10 with the only difference that the reference to Lemma 4.9 should be replaced by a reference to the strengthened Lemma 4.17. □

Completeness of the calculus of L_4 is also a simple extension of the corresponding result for L_3.

Theorem 4.19 (**Completeness of the calculus of L_4**). *The calculus of L_4 is complete.*

Proof. The only thing we need to do is to extend the proof of Theorem 4.16 with two cases: the case where ϕ has the form $a\psi$ and the case where it has the form $\neg a\psi$. The two cases are similar, so we will only consider the case of $a\psi$. So assume that $\sigma a\psi$ occurs on the saturated tableau branch Θ where σ is a nominal urfather. By closure under the rule (@), the branch must also contain formulas τa and $\tau\psi$ for some prefix τ. Urfather Closure (Lemma 4.12) we get $s_\Theta(\tau)\psi \in \Theta$. Using the induction hypothesis this gives us $\mathcal{M}^\Theta, s_\Theta(\tau) \models \psi$. Since $\tau a \in \Theta$ we further get $V^\Theta(a) = \{s_\Theta(\tau)\}$. Thus we have $\mathcal{M}^\Theta, \sigma \models a\psi$, as needed. □

We have now proven soundness, completeness and termination of the calculus of L_4. The termination proof is simple in the sense that tableaus in the calculus are bound to be finite. We do not need a loop-checking procedure in order to ensure termination. However, when we extend the calculus with more of the rules of Figure 2 then termination can no longer be ensured without loop-checks. This is the subject of the following section.

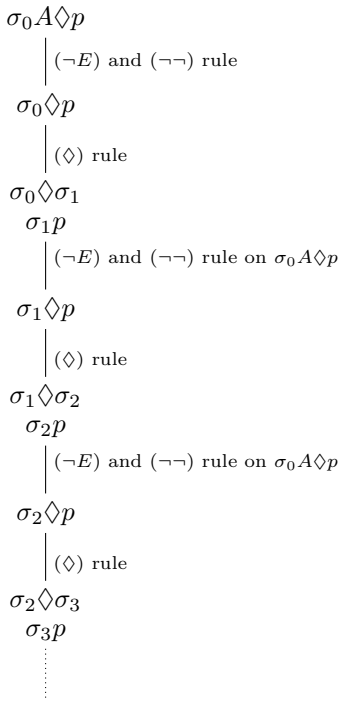


FIGURE 5. An infinite tableau using the $(\neg E)$ rule.

5. Termination with loop-checks

5.1. **Adding the global modality.** We now extend L_4 with the global modality to obtain the following language L_5:

$$(L_5) \qquad \phi ::= p \mid a \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \Diamond_i(\phi_1, \ldots, \phi_{\rho(i)}) \mid a\phi \mid E\phi$$

To obtain a complete calculus for L_5 we need to add the tableau rules (E) and $(\neg E)$. Unfortunately, adding $(\neg E)$ to the calculus creates the same kind of problems as the addition of (Id) did. This is shown by the following example:

Example 5.1 (Non-termination)**.** Consider the L_5 formula $A\Diamond p$, where $\Diamond$ is a unary modal operator. Here we use A as an abbreviation for $\neg E\neg$. Thus a formula $A\phi$ is true at a particular world if and only ϕ is true at all worlds. In the calculus consisting of the rules of L_4 extended with the rule $(\neg E)$ we can then make an infinite tableau with root $\sigma_0 A\Diamond p$, as shown in Figure 5.

The problem shown in the example above is somewhat more serious than the problem of Example 4.7. Let us try to explain why. Call two prefixes σ and τ in a tableau branch *identical worlds* if they make the same set of formulas true on that branch. In order to ensure that tableau construction processes always terminate we need to make sure that identical worlds are either impossible or that we can at least recognise them and take the appropriate actions. Otherwise we might 'reinvent' the same world over and over again in the course of constructing a tableau, and the tableau construction will thus never terminate. In the calculi for L_1 and L_2 one can never construct more than a finite number of identical worlds because of the decreasing length of formulas. In the calculus for L_3 containing the (Id) rule, one can use this rule to construct infinitely many identical worlds as shown in Example 4.7. However, we always have a 'witness' to when two worlds are made identical by the (Id) rule: there is a nominal that the two worlds both make true. Thus we can keep track of identical worlds through the nominals, and this is what allows us to ensure termination by just a simple restriction on the (Id) rule. In the case of the calculus of L_5 containing the $(\neg E)$ rule we can make identical worlds in a similar way as with

the (Id) rule, but we no longer have any witnesses: in order to realise that $\sigma_1, \sigma_2, \ldots$ are identical worlds we need to compare the entire sets of true formulas at those prefixes. This unfortunately means that we need a more heavy-handed method of ensuring ourselves against reinventing the same world over and over again in the course of constructing a tableau—we need *loop-checks*. We will now show how loop-checks can be applied to ensure termination. We will need a stronger notion of urfather that looks at the entire set of true formulas at every prefix.

Definition 5.2. *Let Θ be a branch of a tableau. We define the* inclusion urfather *of a prefix σ on Θ, written $u_\Theta(\sigma)$, to be the earliest introduced prefix τ for which $T^\Theta(\sigma) \subseteq T^\Theta(\tau)$. A prefix σ is called an* inclusion urfather *on Θ if $\sigma = u_\Theta(\tau)$ for some prefix τ.*

Note the similarities between this definition and the definition of nominal urfathers, Definition 4.11. The two notions of urfathers are actually closely related, as the following lemma shows:

Lemma 5.3. *Let Θ be a saturated branch in a calculus containing at least (νId). If σ is a prefix making at least one nominal true on Θ then the nominal urfather and the inclusion urfather of σ coincide.*

Proof. Assume σa occurs on Θ. We need to prove $s_\Theta(\sigma) = u_\Theta(\sigma)$. The nominal urfather of σ is the earliest introduced prefix making some nominal b true that σ also makes true. Since we then have $\sigma b, s_\Theta(\sigma)b \in \Theta$, and since $s_\Theta(\sigma)$ is the earliest prefix making b true, closure under the (νId) rule gives us that all formulas true at σ must also be true at $s_\Theta(\sigma)$. To prove that $s_\Theta(\sigma)$ is the inclusion urfather of σ we then only have to prove that it is the earliest introduced prefix with this property. So assume to obtain a contradiction that there exists a prefix τ introduced earlier than $s_\Theta(\sigma)$ making all formulas true that σ makes true. Then in particular we get τb, contradicting that $s_\Theta(\sigma)$ is the earliest introduced prefix making b true. □

For the nominal urfathers we proved three basic properties in Section 4: Urfather Closure (Lemma 4.12), Urfather Equality (Lemma 4.14), and Urfather Characterisation (Lemma 4.15). By the lemma above these three properties also hold for the new notion of an inclusion urfather, that is, the lemmata still hold when we replace s_Θ by u_Θ and replace 'nominal urfather' by 'inclusion urfather'. Given the assumption of Lemma 5.3, apparently this is only true when considering prefixes making at least one nominal true. However, if a prefix σ is making no nominals true on a branch Θ, then $s_\Theta(\sigma) = \sigma$, so both Urfather Closure and Urfather Characterisation become trivial in this case. Urfather Equality also becomes trivial, since if σ makes no nominals true on Θ, then there doesn't exist any prefixes τ with $\sigma \sim_\Theta \tau$.

Let the *calculus of L_5* be defined to consist of the following rules:

(L_5 rules) $\qquad (\neg), (\neg\neg), (\wedge), (\neg\wedge), (\Diamond), (\neg\Diamond), (Id), (@), (\neg@), (E), (\neg E)$

Note that we have included the unrestricted (Id) rule again. This is because we are now going to ensure termination by a loop-check, and this loop-check will take care of the (Id) rule as well. Our loop-check is formulated as a condition on the construction of tableaus in the calculus of L_5. The condition is as follows:

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

We will first prove that with this restriction in place, termination is again ensured.

Theorem 5.4 (Termination of the calculus of L_5). *Any tableau in the calculus of L_5 constructed under restriction $(\mathcal{R})$ is finite.*

Proof. Assume to obtain a contradiction that there exists a tableau in the calculus containing an infinite branch Θ. Using Lemma 4.3 there must then exist an infinite chain of prefixes

$$\sigma_1 \prec_\Theta \sigma_2 \prec_\Theta \sigma_3 \prec_\Theta \cdots$$

For each $i > 0$ we now define Θ_i to be the initial segment of Θ up to, but not including, the first occurrence of σ_{i+1}. Consider the following sets of formulas:

$$T^{\Theta_1}(\sigma_1), T^{\Theta_2}(\sigma_2), T^{\Theta_3}(\sigma_3), \ldots$$

By the Quasi-subformula Property, Lemma 3.3, these sets are all subsets of the finite set of quasi-subformulas of the root of Θ. Thus there can only be finitely many distinct sets among them, that is, we must have $T^{\Theta_i}(\sigma_i) = T^{\Theta_j}(\sigma_j)$ for some i, j. We can choose i, j such that $i < j$. Then the first occurrence of σ_{i+1} on Θ will be earlier than the first occurrence of σ_{j+1}. Thus Θ_i is an initial segment of Θ_j. Therefore we get $T^{\Theta_i}(\sigma_i) \subseteq T^{\Theta_j}(\sigma_i)$, and since $T^{\Theta_i}(\sigma_i) = T^{\Theta_j}(\sigma_j)$ we thus have

$$T^{\Theta_j}(\sigma_j) \subseteq T^{\Theta_j}(\sigma_i).$$

Since σ_i is introduced earlier on Θ_j than σ_j, this immediately implies that σ_j can not be an inclusion urfather on Θ_j. Now consider the first formula on Θ containing an occurrence of σ_{j+1}. By definition this is the first formula not on Θ_j, and since $\sigma_j \prec_\Theta \sigma_{j+1}$ it must have been introduced by applying one of the prefix generating rules to one of the formulas $\sigma_j\phi$ occurring on Θ_j. This is however in contradiction with restriction $(\mathcal{R})$ since σ_j is not an inclusion urfather on Θ_j. □

The next thing to prove is, as above, completeness. Most of what we need for completeness we have already got. We define for every open, saturated branch Θ a model $\mathcal{M}^\Theta$ similar to the one for the calculus of L_3:

$$\begin{aligned}
\mathcal{M}^\Theta &= (W^\Theta, (R_i^\Theta)_{i<n}, V^\Theta), \text{ where} \\
W^\Theta &= \{\sigma \mid \sigma \text{ is an inclusion urfather on } \Theta\} \\
R_i^\Theta &= \{(\sigma, u_\Theta(\sigma_1), \ldots, u_\Theta(\sigma_{\rho(i)})) \mid \sigma\Diamond_i(\sigma_1, \ldots, \sigma_{\rho(i)}) \text{ occurs on } \Theta \text{ and } \sigma \text{ is an incl. urfather}\} \\
V^\Theta(p) &= \{\sigma \in W^\Theta \mid \sigma p \text{ occurs on } \Theta\} \\
V^\Theta(a) &= \begin{cases} \{\sigma_0\} & \text{if there is no } \sigma \text{ for which } \sigma a \in \Theta. \\ \{u_\Theta(\sigma)\} & \text{if } \sigma a \in \Theta. \end{cases}
\end{aligned}$$

Comparing to the model defined for the calculus of L_3, we have done two things: we have replaced all occurrences of s_Θ by u_Θ and we have restricted the set of worlds to the set of inclusion urfathers on Θ. The first change is simply because we now consider inclusion urfathers instead of nominal urfathers. The second change is needed to ensure completeness when the $(\neg E)$ rule is added, as shown in the proof of completeness below. Note that $V^\Theta(a)$ is still uniquely defined for all nominals a, since as mentioned above we still have the Urfather Equality property (and Θ is closed under applications of (Id) which subsumes (Nom)).

Theorem 5.5 (Completeness of the calculus of L_5). *The calculus of L_5 with restriction $(\mathcal{R})$ is complete.*

Proof. Let Θ be an open, saturated branch in the calculus of L_5 with restriction $(\mathcal{R})$. To prove completeness we will, in similarity with Theorem 4.16, prove the following: for any formula $\sigma\phi$ on Θ where ϕ is an inclusion urfather we have $\mathcal{M}^\Theta, \sigma \models \phi$. The proof is by induction on the syntactic structure of ϕ. The following cases of ϕ were considered in the proof of Theorem 4.16:

$$p, \neg p, a, \neg a, \neg\neg\psi, \psi \wedge \chi, \Diamond_i(\phi_1, \ldots, \phi_{\rho(i)}), \neg\Diamond_i(\phi_1, \ldots, \phi_{\rho(i)})$$

and the cases $a\psi$ and $\neg a\psi$ were considered in Theorem 4.19. The proofs for these cases can all be reused when simply replacing 'nominal urfather' by 'inclusion urfather' and s_Θ by u_Θ. As noted the properties Urfather Closure and Urfather Characterisation used in the proofs still hold when we use the new inclusion urfathers instead of the nominal urfathers. The general point is that whenever we consider a formula $\sigma\phi$ occurring on a saturated branch Θ where σ is an inclusion urfather, then all rules of the calculus have been allowed to be applied to $\sigma\phi$—even the prefix generating ones. The only thing left is thus to consider the cases where ϕ has the form $E\psi$ or $\neg E\psi$. Assume ϕ has the form $E\psi$ and that Θ contains $\sigma E\psi$ where σ is an inclusion urfather. Closure under the (E) rule at urfather prefixes then implies that Θ must also contain a formula $\tau\psi$ for some prefix τ (since σ is an inclusion urfather, application of (E) has not been blocked by restriction $\mathcal{R}$). By Urfather Closure, $u_\Theta(\tau)\psi \in \Theta$. The induction hypothesis then gives us $\mathcal{M}^\Theta, u_\Theta(\tau) \models \psi$ which proves that $\mathcal{M}^\Theta, \sigma \models E\psi$. Assume now that ϕ has the form $\neg E\psi$ and that Θ contains $\sigma\neg E\psi$ where σ is an inclusion urfather. We need to prove $\mathcal{M}^\Theta, \sigma \models \neg E\psi$, that is, for

$\sigma_0(p \wedge A(\Diamond p \wedge \Box^-\Box^-\neg p))$

| $(\wedge)$ rule

$\sigma_0 p$

$\sigma_0 A(\Diamond p \wedge \Box^-\Box^-\neg p)$

| $(\neg E)$ and $(\neg\neg)$ rule

$\sigma_0(\Diamond p \wedge \Box^-\Box^-\neg p)$

| $(\wedge)$ rule

$\sigma_0\Diamond p$

$\sigma_0\Box^-\Box^-\neg p$

| $(\Diamond)$ rule

$\sigma_0\Diamond\sigma_1$

$\sigma_1 p$

| $(\neg E)$ and $(\neg\neg)$ rule on $\sigma_0 A(\Diamond p \wedge \Box^-\Box^-\neg p)$

$\sigma_1(\Diamond p \wedge \Box^-\Box^-\neg p)$

| $(\wedge)$ rule

$\sigma_1\Diamond p$

$\sigma_1\Box^-\Box^-\neg p$

| $(\neg\Diamond^-)$ and $(\neg\neg)$ rule

$\sigma_0\Box^-\neg p$

FIGURE 6. A saturated tableau in the calculus of L with restriction $(\mathcal{R})$.

all $\tau \in W^\Theta$, $\mathcal{M}^\Theta, \tau \models \neg\psi$. Let therefore an arbitrary element τ in W^Θ be chosen. Then τ is an inclusion urfather on Θ. By closure under the $(\neg E)$ rule we must have that $\tau\neg\psi$ occurs on Θ. Since τ is an inclusion urfather, the induction hypothesis gives us $\mathcal{M}^\Theta, \tau \models \neg\psi$, as required. □

5.2. **Adding the inverse modalities.** The final prefixed calculus we are going to consider is obtained by extending the calculus of L_5 with the inverse modalities $\Diamond^-_{i,j}$. This brings us back to the calculus of L containing all of the rules of Figure 2. The inverse modalities pose a problem for our present way of doing things. Let us consider an example.

Example 5.6 (Inverse modalities)**.** Consider the L formula $p \wedge A(\Diamond p \wedge \Box^-\Box^-\neg p)$ where $\Diamond$ is a unary modality. Here we use A as an abbreviation for $\neg E\neg$ and $\Box^-$ as an abbreviation for $\neg\Diamond^-\neg$ (note that since $\Diamond$ is unary it only has a single inverse modality $\Diamond^-$). Under restriction $(\mathcal{R})$ introduced above a saturated tableau with this formula as root will look as in Figure 6. The $(\Diamond)$ rule can not be applied to $\sigma_1\Diamond p$ since σ_0 is the inclusion urfather of σ_1 on the branch, which means that σ_1 is itself not an inclusion urfather and thus the prefix generating rules are blocked at σ_1. However, if we did not have restriction $(\mathcal{R})$ then we could actually make the tableau close by continuing the branch as shown in Figure 7. The extended branch closes since it contains both $\sigma_0 p$ and $\sigma_0\neg p$.

The example shows that we will not get completeness with restriction $(\mathcal{R})$ when we add the inverse modalities. Thus we somehow have to invent a new restriction that will give us completeness without sacrificing termination. We will do this through a third concept of an urfather.

Definition 5.7. *Let Θ be a branch of a tableau. If two prefixes σ and τ make the same set of formulas true on Θ we will call them* twins *on Θ (that is, what we previously called* identical worlds*). A* quasi-urfather *on Θ is a prefix σ for which there are no pair of distinct twins $\tau, \tau' \prec^*_\Theta \sigma$.*

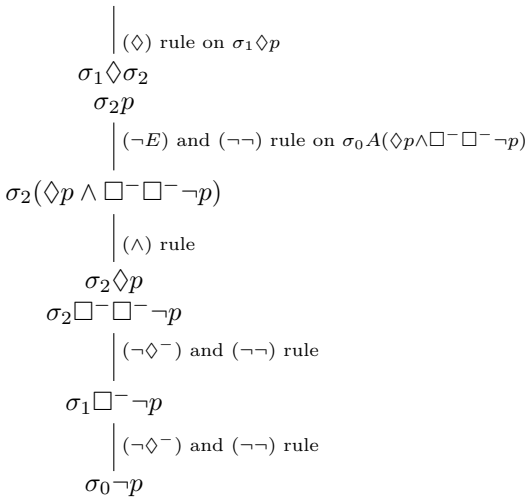


FIGURE 7. Continuation of the tableau of Figure 6 without restriction ($\mathcal{R}$).

Note that if σ is a quasi-urfather on Θ and $\sigma' \prec_\Theta^* \sigma$ then σ' is necessarily also a quasi-urfather. We now define the new restriction on the construction of tableaus, restriction ($\mathcal{D}$), as we defined restriction ($\mathcal{R}$), but with 'inclusion urfather' replaced by 'quasi-urfather':

($\mathcal{D}$) A prefix generating rule is only allowed to be applied to a formula $\sigma\phi$ on a branch if σ is a quasi-urfather on that branch.

We will now prove termination and completeness of the calculus of L with restriction ($\mathcal{D}$).

Theorem 5.8 (**Termination of the calculus of L**). *Any tableau in the calculus of L constructed under restriction ($\mathcal{D}$) is finite.*

Proof. First note that if there exists an infinite tableau with an infinite branch Θ, then by Lemma 4.3 there exists an infinite chain of prefixes

$$\sigma_1 \prec_\Theta \sigma_2 \prec_\Theta \sigma_3 \prec_\Theta \cdots .$$

Let Q be the set of quasi-subformulas of the root formula of Θ, and let n be the cardinality of Q. Let Θ' be the initial segment of Θ up to, but not including, the first occurrence of σ_{2^n+2}. σ_{2^n+2} is then a prefix introduced to Θ by applying a prefix generating rule to a formula of the form $\sigma_{2^n+1}\phi$ on Θ'. Because of restriction ($\mathcal{D}$), σ_{2^n+1} must then be a quasi-urfather on Θ'. However, since all the sets

$$T^{\Theta'}(\sigma_1), T^{\Theta'}(\sigma_2), \ldots, T^{\Theta'}(\sigma_{2^n+1})$$

are subsets of Q, and since Q has cardinality n, at least two of these sets must be identical. This contradicts σ_{2^n+1} being a quasi-urfather on Θ'. □

In proving completeness of the calculus of L things are going to become a bit more complicated than in the previous two cases. In the previous cases we used the urfathers as worlds in the constructed model, and by the definition of urfather there could never be two urfathers making the same nominals true. This does unfortunately not hold for quasi-urfathers, as the following example shows:

Example 5.9 (Quasi-urfathers). Consider the saturated tableau in the calculus of L presented in Figure 8. It consists of a single branch Θ. The branch contains two prefixes σ_1 and σ_1' which both make a true. However, the two prefixes are also both quasi-urfathers. The $\prec_\Theta$ relation looks

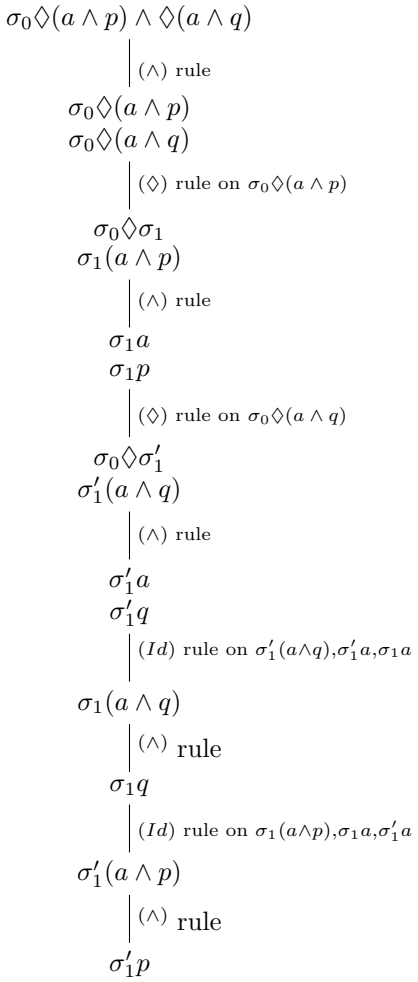
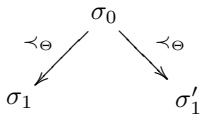


FIGURE 8. A saturated tableau in the calculus of L with restriction ($\mathcal{D}$).

like this:



Since σ_1 and σ'_1 are not related by $\prec_\Theta$, they both become quasi-urfathers on Θ even though they make the same nominals true. Since they make the same nominals true, they must denote the same world.

The example shows that it is possible for two distinct quasi-urfathers to make the same set of nominals true. This was not the case with the two previous concepts of urfathers, as Urfather Equality showed (Lemma 4.14). Since in the model constructed from an open tableau branch we need each nominal to be true in exactly one world, we can not as in the previous cases simply let the worlds of the model be the urfathers. One way to get around this problem would be to build models with equivalence classes of quasi-urfathers as worlds. We will however take another approach which is slightly simpler to present and more in line with the previous completeness proofs: we will define yet another notion of urfather.

Definition 5.10. *Let Θ be a branch of a tableau, and let σ be a prefix occurring on Θ. The* identity urfather *of σ on Θ, written $v_\Theta(\sigma)$, is the earliest introduced prefix τ satisfying:*

(1) τ *is a twin of* σ.
(2) τ *is a quasi-urfather.*

If such a prefix does not exist we let $v_\Theta(\sigma)$ *be undefined. Thus* v_Θ *is only a partially defined mapping. A prefix* σ *is called an* identity urfather *on* Θ *if* $\sigma = v_\Theta(\tau)$ *for some prefix* τ.

Example 5.11 (Identity Urfathers)**.** Consider again the branch Θ presented in Figure 8. As noted in Example 5.9, both σ_1 and σ_1' are quasi-urfathers. We also see that $T^\Theta(\sigma_1) = T^\Theta(\sigma_1')$, so σ_1 and σ_1' must be twins. Since σ_1 is introduced earlier to Θ than σ_1', σ_1 is the identity urfather of σ_1'. Thus the only identity urfathers on Θ are σ_1 and the root prefix σ_0. We can of course build a model of the root formula of Θ by using these two prefixes as the set of worlds.

Note that if σ is a quasi-urfather on a branch Θ then $v_\Theta(\sigma)$ is necessarily defined. Note also that the root prefix of a branch Θ will always be an identity urfather on that branch. We will use standard notation and express that $v_\Theta(\sigma)$ is defined by writing $\sigma \in \mathrm{dom}(v_\Theta)$. We have the following result:

Lemma 5.12. *Let* Θ *be a branch of a tableau, and let* σ *be a quasi-urfather on* Θ*. If* $\sigma \prec_\Theta \tau$ *then* $\tau \in \mathrm{dom}(v_\Theta)$.

Proof. Assume $\sigma \prec_\Theta \tau$ where σ is a quasi-urfather on Θ. We need to prove $\tau \in \mathrm{dom}(v_\Theta)$. If τ is a quasi-urfather on Θ then this is trivial. So assume conversely that τ is not a quasi-urfather. Then there must exist a pair of distinct twins γ, γ' with $\gamma \prec_\Theta^* \gamma' \prec_\Theta^* \tau$. Since σ is a quasi-urfather we can not have both $\gamma \prec_\Theta^* \sigma$ and $\gamma' \prec_\Theta^* \sigma$. Since $\sigma \prec_\Theta \tau$ this implies $\gamma' = \tau$, using Lemma 4.2. Thus τ has γ as a twin, and since necessarily $\gamma \prec_\Theta^* \sigma$ we get that γ is a quasi-urfather. Since τ thus has a quasi-urfather twin, $v_\Theta(\tau)$ must necessarily be defined. □

As in the two previous cases we also have Urfather Closure, Urfather Equality, and Urfather Characterisation results.

Lemma 5.13 (**Urfather Closure**)**.** *Let* Θ *be a branch of a tableau in any calculus. If* $\sigma\phi$ *occurs on* Θ *and* $\sigma \in \mathrm{dom}(v_\Theta)$ *then* $v_\Theta(\sigma)\phi$ *also occurs on* Θ.

Proof. Since $v_\Theta(\sigma)$ by definition is a twin of σ, the two prefixes make the same formulas true on Θ. That is, if $\sigma\phi$ occurs on Θ then so does $v_\Theta(\sigma)\phi$. □

Lemma 5.14 (**Urfather Equality**)**.** *Let* Θ *be a saturated branch in the calculus of* L*. If* σ *and* τ *are two elements of* $\mathrm{dom}(v_\Theta)$ *both making some nominal* a *true on* Θ*, then* $v_\Theta(\sigma) = v_\Theta(\tau)$.

Proof. Assume $\sigma, \tau \in \mathrm{dom}(v_\Theta)$ and $\sigma a, \tau a \in \Theta$. Since Θ is saturated, closure under the (Id) rule implies that σ and τ must make the same set of formulas true. Thus they are twins. The definition of v_Θ then immediately implies $v_\Theta(\sigma) = v_\Theta(\tau)$. □

Lemma 5.15 (**Urfather Characterisation**)**.** *Let* Θ *be a branch of a tableau in any calculus. Then* σ *is an identity urfather if and only if* $v_\Theta(\sigma) = \sigma$.

Proof. Both the 'if' and 'only if' direction immediately follows from the definitions of v_Θ and 'identity urfather'. □

We are now ready for the model construction. Given an open, saturated branch Θ with root $\sigma_0\phi_0$ in the calculus of L, we define a model $\mathcal{M}^\Theta$ by

$$
\begin{aligned}
\mathcal{M}^\Theta &= (W^\Theta, (R_i^\Theta)_{i<n}, V^\Theta), \text{ where} \\
W^\Theta &= \{\sigma \mid \sigma \text{ is an identity urfather on } \Theta\} \\
R_i^\Theta &= \{(v_\Theta(\sigma), v_\Theta(\sigma_1), \ldots, v_\Theta(\sigma_{\rho(i)})) \mid \sigma\Diamond_i(\sigma_1, \ldots, \sigma_{\rho(i)}) \text{ occurs on } \Theta \text{ and} \\
&\qquad\qquad \sigma, \sigma_1, \ldots, \sigma_{\rho(i)} \in \mathrm{dom}(v_\Theta)\} \\
V^\Theta(p) &= \{\sigma \in W^\Theta \mid \sigma p \text{ occurs on } \Theta\} \\
V^\Theta(a) &= \begin{cases} \{\sigma_0\} & \text{if there is no } \sigma \in \mathrm{dom}(v_\Theta) \text{ for which } \sigma a \in \Theta. \\ \{v_\Theta(\sigma)\} & \text{if } \sigma a \in \Theta \text{ where } \sigma \in \mathrm{dom}(v_\Theta). \end{cases}
\end{aligned}
$$

That $V^\Theta(a)$ is uniquely defined for any nominal a follows directly from Urfather Equality (Lemma 5.14). We are now finally ready for the completeness proof of L with restriction $(\mathcal{D})$.

Theorem 5.16 (Completeness of the calculus of L). *Let Θ be an open, saturated branch in the calculus of L with restriction $(\mathcal{D})$. For any formula $\sigma\phi \in \Theta$ where σ is an identity urfather we have $\mathcal{M}^\Theta, \sigma \models \phi$. In particular, we have $\mathcal{M}^\Theta, \sigma_0 \models \phi_0$ where $\sigma_0\phi_0$ is the root of the tableau.*

Proof. As in the previous completeness proofs, the proof is by induction on the syntactic structure of ϕ. In the cases where ϕ has one of the forms p, $\neg p$, a, $\neg\neg\psi$, $\psi \wedge \chi$, $\neg(\psi \wedge \chi)$, $\neg\Diamond_i(\phi_1, \ldots, \phi_{\rho(i)})$ or $\neg E\psi$ we can directly reuse the previously given proofs by simply replacing references to s_Θ and u_Θ by v_Θ and replacing references to 'nominal urfather' and 'inclusion urfather' by 'identity urfather'. This is because we still have the basic properties Urfather Closure (Lemma 5.13) and Urfather Characterisation (Lemma 5.15). If ϕ has one of the forms $\neg a$, $\Diamond_i(\phi_1, \ldots, \phi_{\rho(i)})$, $a\psi$, $\neg a\psi$ or $E\psi$ then we can also reuse the previously given proofs, but in all cases we need to add the following small piece of argumentation: when a prefix generating rule is applied to a premise $\sigma\psi$ to produce a conclusion $\tau\chi$, and when furthermore σ is an identity urfather, then $\tau \in \mathrm{dom}(v_\Theta)$. This is a direct consequence of Lemma 5.12. This piece of argumentation is needed since now the urfather mapping v_Θ is only partially defined, and we need to ensure that we only apply it to prefixes for which it is defined. The only remaining cases of ϕ to consider are $\Diamond^-_{i,j}(\phi_1, \ldots, \phi_{\rho(i)})$ and $\neg\Diamond^-_{i,j}(\phi_1, \ldots, \phi_{\rho(i)})$. First assume Θ contains $\sigma\Diamond^-_{i,j}(\phi_1, \ldots, \phi_{\rho(i)})$ where σ is an identity urfather. Then by closure under the rule $(\Diamond^-)$ at identity urfather prefixes we get that for some prefixes $\sigma_1, \ldots, \sigma_{\rho(i)}$, Θ contains $\sigma_1\Diamond_i(\sigma_2, \ldots, \sigma_j, \sigma, \sigma_{j+1}, \ldots, \sigma_{\rho(i)})$ as well as $\sigma_k\phi_k$ for all $k = 1, \ldots, \rho(i)$. The prefixes $\sigma_1, \ldots, \sigma_{\rho(i)}$ are generated by this particular rule application, so we have $\sigma \prec_\Theta \sigma_k$ for all $k = 1, \ldots, \rho(i)$. Since σ is an identity urfather, Lemma 5.12 gives us $\sigma_k \in \mathrm{dom}(v_\Theta)$ for all $k = 1, \ldots, \rho(i)$. Thus by definition of R_i^Θ, we get

$$(v_\Theta(\sigma_1), v_\Theta(\sigma_2), \ldots, v_\Theta(\sigma_j), v_\Theta(\sigma), v_\Theta(\sigma_{j+1}), \ldots, v_\Theta(\sigma_{\rho(i)})) \in R_i^\Theta$$

which, by Urfather Characterisation, is equivalent to

$$(v_\Theta(\sigma_1), v_\Theta(\sigma_2), \ldots, v_\Theta(\sigma_j), \sigma, v_\Theta(\sigma_{j+1}), \ldots, v_\Theta(\sigma_{\rho(i)})) \in R_i^\Theta \tag{4}$$

By Urfather Closure, Θ contains $v_\Theta(\sigma_k)\phi_k$ for all $k = 1, \ldots, \rho(i)$. The induction hypothesis then gives $\mathcal{M}^\Theta, v_\Theta(\sigma_k) \models \phi_k$ for all $k = 1, \ldots, \rho(i)$. Combining this with (4) finally gives us $\mathcal{M}^\Theta, \sigma \models \Diamond^-_{i,j}(\phi_1, \ldots, \phi_{\rho(i)})$, as needed. Consider the case where ϕ has the form $\neg\Diamond^-_{i,j}(\phi_1, \ldots, \phi_{\rho(i)})$. Assume Θ contains $\sigma\neg\Diamond^-_{i,j}(\phi_1, \ldots, \phi_{\rho(i)})$ where σ is an identity urfather. We need to prove $\mathcal{M}^\Theta, \sigma \models \neg\Diamond^-_{i,j}(\phi_1, \ldots, \phi_{\rho(i)})$. If there do not exist worlds $\sigma_1, \ldots, \sigma_{\rho(i)}$ such that

$$(\sigma_1, \ldots, \sigma_j, \sigma, \sigma_{j+1}, \ldots, \sigma_{\rho(i)}) \in R_i^\Theta$$

then this holds trivially. Otherwise, let such $\sigma_1, \ldots, \sigma_{\rho(i)}$ be chosen arbitrarily. We then need to prove the existence of an $l \in \{1, \ldots, \rho(i)\}$ such that $\mathcal{M}^\Theta, \sigma_l \models \neg\phi_l$. By definition of R_i^Θ there must exist prefixes $\sigma'_1, \ldots, \sigma'_{\rho(i)}$ such that $\sigma_k = v_\Theta(\sigma'_k)$ for all $k = 1, \ldots, \rho(i)$ and Θ contains $\sigma'_1\Diamond_i(\sigma'_2, \ldots, \sigma'_j, \sigma, \sigma'_{j+1}, \ldots, \sigma'_{\rho(i)})$. Since Θ is saturated, it must then also contain $\sigma'_l\neg\phi_l$ for some $l \in \{1, \ldots, \rho(i)\}$ (closure under the $(\neg\Diamond^-)$ rule). Urfather Closure then gives $v_\Theta(\sigma'_l)\neg\phi_l \in \Theta$, that is, $\sigma_l\neg\phi_l \in \Theta$. Since σ_l is an identity urfather, the induction hypothesis gives $\mathcal{M}^\Theta, \sigma_l \models \neg\phi_l$, as required. □

We can extend the language L further to include the down-arrow binder and/or quantifiers over nominals. However, the corresponding logics are known to be undecidable [2], so we have no way of making terminating tableau calculi for these extended logics. However, we can still prove completeness. It is simple to extend the tableau calculus of L to a complete calculus for the language including both the down-arrow binder and quantifiers over nominals; for details, see [8] or [3].

6. Internalised tableau calculi

So far we have only considered prefixed tableau calculi. But it is also possible to give fully *internalised* tableau calculi; that is, calculi in which all tableau formulas belong to the object language. Such tableaus are of theoretical interest because when they are extended with pure axioms (that is, axioms whose only atoms are nominals) they are automatically complete with respect to the class of frames the axioms define. Nonetheless, they seem more complex than prefixed systems, and at present the computational significance of pure axioms is not well understood (adding pure axioms can easily lead to non-terminating behaviour, as we discuss at the end of the paper), so prefixed systems seem to be regarded as the more down-to-earth option. Somewhat to our surprise, however, it turns out that it is possible to define an internalised tableau system that terminates without loop-checks and without the need for non-local side conditions on the rules. That is, the tableau system we are about to discuss, although internalised, is the simplest one we know of for hybrid logic.

In a prefixed tableau systems, each formula is either a prefixed formula of the form $\sigma\phi$ or an accessibility formula of the form $\sigma\Diamond_i(\sigma_1,\ldots,\sigma_{\rho(i)})$. Both prefixed formulas and accessibility formulas are meta-formulas: they contain prefixes, which are meta-linguistic symbols. In an internalised tableau calculus all formulas in a tableau need to be pure object language formulas, that is, not containing any meta-linguistic symbols like prefixes. In the case of hybrid logic it is not difficult to turn a prefixed tableau calculus into an internalised one. We simply perform the following trick: we turn the prefixes into nominals, that is, we replace the requirement $\mathsf{Pref}\cap\mathsf{Nom}=\emptyset$ by the requirement $\mathsf{Pref}\subseteq\mathsf{Nom}$. Then all the formulas occurring in the tableau rules of Figure 2 become object language formulas, in fact they all become satisfaction statements in hybrid logic. Thus the calculus becomes internalised.

Since the only thing we have done to internalise the prefixed calculus is to count the prefixes among the nominals, it is not hard to prove that all the previous tableau-based decidability results carry over to the internalised framework—we simply reuse all the previously given proofs. That is, we automatically get internalised calculi for all of the considered languages $L_1, L_2, \ldots, L_5$ and L—and in addition we get termination results for them for free. However, a little caution is needed. If we want to reuse the previous proofs, then we also need to retain an explicit distinction between the nominals that belong to Pref and those which doesn't. The nominals in Pref are those which are denoted $\sigma,\tau,\ldots$ in the tableau rules, and the nominals in $(\mathsf{Nom}-\mathsf{Pref})$ are those which are denoted $a,b,\ldots$ in the tableau rules. Distinguishing explicitly between two sorts of nominals in ones tableau rules is of course not particularly elegant. Fortunately, it is possible to do away with this and retain the tableau-based decision algorithms in only slightly modified form. However, it appears that even if we have only one sort of nominals in the tableau rules, we still need some distinction between the nominals that 'act as prefixes' and the nominals that don't. Consider for instance the internalised version of the $(\Diamond)$ rule, which we denote $[\Diamond]$,

$$\frac{a\Diamond_i(\phi_1,\ldots,\phi_{\rho(i)})}{\begin{array}{c}a\Diamond_i(a_1,\ldots,a_{\rho(i)})\\ a_1\phi_1\\ \vdots\\ a_{\rho(i)}\phi_{\rho(i)}\end{array}}\,[\Diamond]$$

Consider the first conclusion of this rule, $a\Diamond_i(a_1,\ldots,a_{\rho(i)})$. This is obviously a formula corresponding to an accessibility formula in the prefixed calculus. So the nominals $a_1,\ldots,a_{\rho(i)}$ act as prefixes. In order to ensure a terminating tableau procedure it is necessary to be able to distinguish between occurrences of formulas of the form $a\Diamond_i(a_1,\ldots,a_{\rho(i)})$ where the a_j act as prefixes and occurrences where they do not. This can be done by introducing the following simple definition. A formula of the form $a\Diamond_i(a_1,\ldots,a_{\rho(i)})$ occurring in an tableau is called an *accessibility formula* if it is the first conclusion of an application of $[\Diamond]$. As we will see below, it is sufficient to be able to distinguish between accessibility formulas and other formulas.

The internalisation obtained by simply letting $\mathsf{Pref} \subseteq \mathsf{Nom}$ does not in itself give us much new. The real advantage of the internalisation is the fact that we can now simplify the rules of Figure 2 to obtain an internalised calculus which is simpler than what we have been able to obtain in the prefixed case. If one looks for simplifications in the internalised version of the rules of Figure 2, it is immediate that (@) and ($\neg$@) are more complicated than they need to be. In (@), we don't really need a new nominal τ to witness the fact that a and ϕ hold at the same world. We can simply replace the rule by the following one:

$$\frac{ab\phi}{b\phi}\,[@]$$

Similarly, we can simplify ($\neg$@) to:

$$\frac{a\neg b\phi}{b\neg\phi}\,[\neg@]$$

Finally, and most importantly, we can simplify the (Id) rule. It can be replaced by the following:

$$\frac{a\phi, ab}{b\phi}\,[Id]$$

At first, this might not appear to be a significant simplification. But it is: with this $[Id]$ rule instead of the original prefixed one we can give a termination proof of an internalised calculus for L_4 in which neither loop-checks nor non-local side conditions on the tableau rules are needed. Compare this with the prefixed calculus of L_4, where we had to ensure termination by replacing the original (Id) rule by the two rules, (Nom) and (νId), where (νId) had a non-trivial and non-local side condition (a side condition in which the entire branch had to be taken into account). In the internalised system, on the other hand, the only side condition we need to ensure termination of the internalised calculus is that $[Id]$ is not applied to accessibility formulas. Before we show how termination is obtained, let us present the entire internalised calculus for the language L_4. Recall that L_4 is the language given by the following grammar.

$$(L_4) \qquad \phi \ ::= \ p \mid a \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \Diamond_i(\phi_1, \ldots, \phi_{\rho(i)}) \mid a\phi$$

The internalised calculus for L_4 is presented in Figure 9. Our goal in the following is to show that the calculus of Figure 9 is both terminating and complete—using no loop-checks. Thus it gives rise to a much nicer decision procedure than the prefixed one for L_4. Furthermore, the calculus of Figure 9 also provides us with a simplification of the original internalised calculus of Blackburn [3]. Blackburn's system contains rules corresponding to all of the rules of Figure 9, but it also contains some additional ones: [Ref], [Sym], and [Bridge]. These additional rules actually make the calculus non-terminating, so a loop-check is needed to ensure termination (this is how termination is ensured in the tableau-based decision procedure for Blackburn's system given in [5]). However, in our simplified calculus, termination is ensured without any loop-checks, as shown below. So not only are the rules [Ref], [Sym], and [Bridge] not needed for completeness, by leaving them out we can actually make a much simpler decision method for the logic.

In the following we will refer to the calculus consisting of the rules of Figure 9 as the *internalised calculus*. Tableaus in this calculus are defined using the same conventions as for the prefixed calculus. Most important, a tableau branch is said to be *closed* if it contains a pair of formulas $a\phi$ and $a\neg\phi$, where a is a nominal and ϕ is any formula. Otherwise it is called *open*. A tableau in the internalised calculus will be called an *internalised tableau*. Let $\mathcal{T}$ be an internalised tableau. If a is a nominal occurring in the root formula of the tableau then a is called a *root nominal* of $\mathcal{T}$. Other nominals occurring on $\mathcal{T}$ are called *non-root nominals* of $\mathcal{T}$. Let Θ be a branch of an internalised tableau. If a rule application on Θ has a formula $a\phi$ as one of its conclusions, then $a\phi$ is said to be *produced* by that rule on Θ. For convenience, the notion of quasi-subformulas is redefined in the following way. We say that a formula $a\phi$ is a *quasi-subformula* of a formula $b\psi$ if either ϕ is a subformula of ψ or ϕ has the form $\neg\chi$, where χ is a subformula of ψ.

$$\frac{a\neg b}{bb}\,[\neg] \qquad \frac{a\neg\neg\phi}{a\phi}\,[\neg\neg]$$

$$\frac{a(\phi\wedge\psi)}{\begin{array}{c}a\phi\\ a\psi\end{array}}\,[\wedge] \qquad \frac{a\neg(\phi\wedge\psi)}{a\neg\phi \mid a\neg\psi}\,[\neg\wedge]$$

$$\frac{a\Diamond_i(\phi_1,\ldots,\phi_{\rho(i)})}{\begin{array}{c}a\Diamond_i(a_1,\ldots,a_{\rho(i)})\\ a_1\phi_1\\ \vdots\\ a_{\rho(i)}\phi_{\rho(i)}\end{array}}\,[\Diamond]^{1,2} \qquad \frac{\begin{array}{c}a\neg\Diamond_i(\phi_1,\ldots,\phi_{\rho(i)})\\ a\Diamond_i(a_1,\ldots,a_{\rho(i)})\end{array}}{a_1\neg\phi_1 \mid a_2\neg\phi_2 \mid \cdots \mid a_{\rho(i)}\neg\phi_{\rho(i)}}\,[\neg\Diamond]$$

$$\frac{ab\phi}{b\phi}\,[@] \qquad \frac{a\neg b\phi}{b\neg\phi}\,[\neg@]$$

$$\frac{a\phi, ab}{b\phi}\,[Id]^2$$

[1] The nominals $a_1,\ldots,a_{\rho(i)}$ are all new to the tableau.
[2] None of the premises are accessibility formulas.

FIGURE 9. Internalised calculus for the hybrid language L_4.

6.1. **Termination of the internalised calculus.** In proving termination of the internalised calculus, we will follow the termination proofs given for the prefixed calculi very closely. First we have a quasi-subformula property corresponding to Lemma 3.3.

Lemma 6.1 (**Quasi-subformula Property**). *Let $\mathcal{T}$ be an internalised tableau. For any formula $a\phi$ occurring on $\mathcal{T}$, one of the following holds:*

- *$a\phi$ is a quasi-subformula of the root formula of $\mathcal{T}$.*
- *$a\phi$ is an accessibility formula of $\mathcal{T}$.*

Proof. As for Lemma 3.3, this is easily proven by going through each of the rules of the tableau calculus. □

Let Θ be a branch of an internalised tableau. For each nominal a occurring on Θ we define a set of formulas $T^{\Theta}(a)$ by

$$T^{\Theta}(a) = \{\phi \mid a\phi \in \Theta \text{ and } a\phi \text{ is a quasi-subformula of the root formula}\}.$$

We now have the following result, corresponding to Lemma 3.4.

Lemma 6.2. *Let Θ be a branch of an internalised tableau, and let a be any nominal occurring on Θ. The set $T^{\Theta}(a)$ is finite.*

Proof. Trivial from the definition of $T^{\Theta}(a)$. □

Note that in the internalised calculus the only rule that can introduce new nominals to a tableau is the $[\Diamond]$ rule. The following definition corresponds to Definition 4.1.

Definition 6.3. *Let Θ be a branch of an internalised tableau. If a nominal b has been introduced to the branch by applying the rule $[\Diamond]$ to a premise $a\Diamond_i(\phi_1,\ldots,\phi_{\rho(i)})$ then we say that b is* generated *by a on Θ, and we write $a \prec_{\Theta} b$.*

Note that when a nominal b has been introduced to a branch by an application of $[\Diamond]$ to a premise $a\Diamond_i(\phi_1, \dots, \phi_{\rho(i)})$ then the first conclusion of that rule application will be an accessibility formula of the form $a\Diamond_i(\dots, b, \dots)$. Thus a nominal b is generated by a nominal a on a branch if and only if that branch contains an accessibility formula of the form $a\Diamond_i(\dots, b, \dots)$.

We now have the following lemma, corresponding to Lemma 4.2.

Lemma 6.4. *Let Θ be a branch of an internalised tableau. The graph $G = (N^\Theta, \prec_\Theta)$, where N^Θ is the set of nominals occurring on Θ, is a finite set of wellfounded, finitely branching trees.*

Proof. That G is wellfounded and finitely branching is proved exactly as in Lemma 4.2, except that reference to Lemma 3.4 is replaced by reference to the variant for the internalised calculus, Lemma 6.2. What is left is to prove that G is a finite set of trees. This follows from the fact that each nominal in N^Θ can be generated by at most one other nominal, and the fact that each nominal in N^Θ must have one of the finitely many root nominals as an ancestor. □

Lemma 6.5. *Let Θ be a branch of an internalised tableau. Then Θ is infinite if and only if there exists an infinite chain of nominals*

$$a_1 \prec_\Theta a_2 \prec_\Theta a_3 \prec_\Theta \cdots .$$

Proof. The proof is similar to the proof of Lemma 4.3. The 'if' direction is trivial. To prove the 'only if' direction, let Θ be any infinite tableau branch. Θ must contain infinitely many distinct nominals, since it follows immediately from Lemma 6.1 that a tableau with finitely many nominals can only contain finitely many distinct formulas. This implies that the graph $G = (N^\Theta, \prec_\Theta)$ defined as in Lemma 6.4 must be infinite. Since by Lemma 6.4, G is a finite set of wellfounded, finitely branching trees, G must then contain an infinite path $(a_1, a_2, a_3, \dots)$. Thus we get an infinite chain $a_1 \prec_\Theta a_2 \prec_\Theta a_3 \prec_\Theta \cdots$. □

Definition 6.6. *Let Θ be a branch of an internalised tableau, and let a be a nominal occurring on Θ. We define $m_\Theta(a)$ by*

$$m_\Theta(a) = \max\{|\phi| \mid a\phi \in \Theta\}.$$

Corresponding to Lemma 4.17 we now have the following result.

Lemma 6.7 (Decreasing length). *Let Θ be a branch of an internalised tableau. If $a \prec_\Theta b$ then $m_\Theta(a) > m_\Theta(b)$.*

Proof. Assume $a \prec_\Theta b$. Let ϕ be a formula of maximal length true at b on Θ. We need to prove $m_\Theta(a) > |\phi|$. The formula $b\phi$ can not have been introduced on Θ by applying any of the rules $[\neg\neg]$, $[\wedge]$ or $[\neg\wedge]$, since this contradicts the maximality of ϕ. Assume $b\phi$ has been introduced by applying $[@]$ to a premise of the form $ab\phi$. By Lemma 6.1, $ab\phi$ is a quasi-subformula of the root formula. Thus b must be a root nominal. However, this is a contradiction, since by assumption b is generated by a, and can thus not be a root nominal. Thus $[@]$ can not have been the rule producing $b\phi$. A similar argument shows that $b\phi$ can not have been produced by $[\neg@]$ either. Now assume that $b\phi$ has been introduced by applying $[Id]$ to premises $a\phi$ and ab. Then by Lemma 6.1, ab is a quasi-subformula of the root formula, and thus b is again a root nominal, which is a contradiction. Thus $b\phi$ can not have been produced by $[Id]$ either. Thus $b\phi$ must have been introduced by one of the rules $[\neg]$, $[\Diamond]$ or $[\neg\Diamond]$. In the case of $[\neg]$ we get $|\phi| = 1$. Since a has generated b, there must be at least one formula ψ of length > 1 such that $a\psi \in \Theta$. Therefore, $m_\Theta(a) > |\phi|$, as needed. In the case of $[\Diamond]$, the rule instance producing $b\phi$ must have the following form

$$\dfrac{a'\Diamond_i(\dots, \phi, \dots)}{\begin{array}{c} a'\Diamond_i(\dots, b, \dots) \\ \vdots \\ b\phi \\ \vdots \end{array}}$$

Note that $b\phi$ can never be the first conclusion of this rule instance, since that would contradict the maximality of ϕ. By definition of $\prec_\Theta$ we must now have $a' \prec_\Theta b$. Since we have already assumed $a \prec_\Theta b$, Lemma 6.4 implies $a' = a$. Thus we get

$$m_\Theta(a) = m_\Theta(a') \geq |\Diamond_i(\dots, \phi, \dots)| > |\phi|\,,$$

as needed. Now consider the case where $b\phi$ is introduced by an application of $[\neg\Diamond]$. In this case $b\phi$ must be on the form $b\neg\psi$ and introduced by applying $[\neg\Diamond]$ to a pair of premises of the form

$$a'\neg\Diamond_i(\dots, \psi, \dots),\ a'\Diamond_i(\dots, b, \dots).$$

Consider the premise $a'\Diamond_i(\dots, b, \dots)$. Since b is a non-root nominal, the formula $a'\Diamond_i(\dots, b, \dots)$ can not be a quasi-subformula of the root formula. Thus, by Lemma 6.1, the formula must be an accessibility formula. This implies $a' \prec_\Theta b$, and thus Lemma 6.4 again implies $a' = a$. Therefore we get

$$m_\Theta(a) = m_\Theta(a') \geq |\neg\Diamond_i(\dots, \psi, \dots)| > |\neg\psi| = |\phi|\,,$$

as required. □

We can now finally prove termination of the internalised calculus. We simply use the same argument as in the proof of termination for the calculus of L_2 (Theorem 4.6).

Theorem 6.8 (**Termination of the internalised calculus).** *Any tableau in the internalised calculus is finite.*

Proof. Assume there exists an infinite internalised tableau. Then it must have an infinite branch Θ. By Lemma 6.5, there exists an infinite chain

$$a_1 \prec_\Theta a_2 \prec_\Theta a_3 \prec_\Theta \cdots.$$

Now by Lemma 6.7 we have

$$m_\Theta(a_1) > m_\Theta(a_2) > m_\Theta(a_3) > \cdots$$

which is a contradiction, since $m_\Theta(a)$ is a non-negative number for any nominal a. □

6.2. **Completeness of the internalised calculus.** We will now prove completeness of the internalised calculus. The completeness proof closely follows the lines of the previously given completeness proofs for the prefixed calculi. Let Θ be a branch of an internalised tableau, and let a be a nominal occurring on Θ. a is called a *right nominal* on Θ if there exists a nominal b such that the formula ba occurs on Θ. Note that by Lemma 6.1, all right nominals must be root nominals.

Definition 6.9. *Let Θ be a branch of an internalised tableau. We define a binary relation $\sim_\Theta$ on the right nominals of Θ by*

$$a \sim_\Theta b \quad \textit{iff} \quad ab \in \Theta.$$

Lemma 6.10. *Let Θ be a saturated branch of an internalised tableau. The relation $\sim_\Theta$ is an equivalence relation.*

Proof. We need to prove that $\sim_\Theta$ is reflexive, symmetric, and transitive. To prove reflexivity, let a be any right nominal of Θ. Then, by definition, there must be a nominal b such that ba occurs on Θ. Now, if we apply $[Id]$ with the premise ba in both positions then we get the conclusion aa. Since Θ is saturated it is in particular closed under all possible applications of $[Id]$, and thus aa must occur on Θ. This proves reflexivity. To prove symmetry, assume $ab \in \Theta$, where a and b are right nominals. We need to prove $ba \in \Theta$. By reflexivity, we have $aa \in \Theta$. Applying $[Id]$ to premises aa and ab gives the conclusion ba. Since Θ is saturated this implies $ba \in \Theta$, proving symmetry. To prove transitivity, assume $ab, bc \in \Theta$ where a, b and c are right nominals. By symmetry we have $ba \in \Theta$. Applying $[Id]$ with premises bc and ba gives the conclusion ac, as needed. □

Definition 6.11. *Let Θ be a saturated branch of an internalised tableau, and let a be a right nominal of Θ. The equivalence class of a with respect to $\sim_\Theta$ is denoted $[a]_\Theta$.*

We can assume that the set of nominals Nom is totally ordered, for instance by simply letting $\mathsf{Nom} = \mathbb{N}$, and use the standard ordering on $\mathbb{N}$. Given such a fixed total order on the nominals, we will for any finite set of nominals A use $\min(A)$ to denote the smallest member of A with respect to that ordering.

Definition 6.12 (**Urfathers**). *Let Θ be a saturated branch of an internalised tableau, and let a be any nominal occurring on Θ. The* urfather *of a on Θ, denoted $u_\Theta(a)$, is defined by*

$$u_\Theta(a) = \begin{cases} \min([b]_\Theta) & \text{if } ab \in \Theta \\ a & \text{otherwise.} \end{cases}$$

We need to ensure that u_Θ is uniquely defined by the above cases. It suffices to prove that if ab and ac both occur on Θ then $[b]_\Theta = [c]_\Theta$. So assume $ab, ac \in \Theta$. Since $[Id]$ can be applied to these two premises to give the conclusion cb, we must have $cb \in \Theta$, since Θ is saturated. This then implies $c \sim_\Theta b$ and thus $[c]_\Theta = [b]_\Theta$, as needed. We have the following simple result concerning urfathers.

Lemma 6.13. *Let Θ be a saturated branch of an internalised tableau. If a is a right nominal of Θ then $u_\Theta(a) = \min([a]_\Theta)$.*

Proof. If a is a right nominal then by Lemma 6.10, $aa \in \Theta$. Using the definition of u_Θ this immediately implies $u_\Theta(a) = \min([a]_\Theta)$. □

For completeness, we are going to need one of our old friends, Urfather Closure.

Lemma 6.14 (**Urfather Closure**). *Let Θ be a saturated branch of an internalised tableau. If $a\phi \in \Theta$ and $a\phi$ is not an accessibility formula then $u_\Theta(a)\phi \in \Theta$.*

Proof. Assume $a\phi \in \Theta$, where $a\phi$ is not an accessibility formula. If $u_\Theta(a) = a$ then there is nothing to prove. So assume $u_\Theta(a) = \min([b]_\Theta)$ where $ab \in \Theta$. Applying $[Id]$ to premises $a\phi, ab$ gives the conclusion $b\phi$. Since $u_\Theta(a) = \min([b]_\Theta)$ we must have $b \sim_\Theta u_\Theta(a)$ and thus $bu_\Theta(a) \in \Theta$. Applying $[Id]$ to premises $b\phi, bu_\Theta(a)$ gives the conclusion $u_\Theta(a)\phi$. Since Θ is saturated, it is in particular closed under all possible applications of $[Id]$ where the premises are not accessibility formulas, and thus we must have $u_\Theta(a)\phi \in \Theta$, as needed. □

Given an open, saturated branch Θ of an internalised tableau, we now define a model $\mathcal{M}^\Theta$ by

$$\begin{aligned} \mathcal{M}^\Theta &= (W^\Theta, (R_i^\Theta)_{i<n}, V^\Theta), \text{ where} \\ W^\Theta &= \{a \mid a \text{ is a nominal occurring on } \Theta\} \\ R_i^\Theta &= \{(a, u_\Theta(a_1), \ldots, u_\Theta(a_{\rho(i)})) \mid a\Diamond_i(a_1, \ldots, a_{\rho(i)}) \text{ occurs on } \Theta\} \\ V^\Theta(p) &= \{a \mid ap \text{ occurs on } \Theta\} \\ V^\Theta(a) &= \{u_\Theta(a)\}. \end{aligned}$$

Note that we have a simpler definition of V^Θ than for the prefixed calculus. We are now ready to prove completeness.

Theorem 6.15 (**Completeness**). *Let Θ be an open, saturated branch of an internalised tableau, and let $a\phi$ be a formula occurring on Θ which is not an accessibility formula. Then $\mathcal{M}^\Theta, u_\Theta(a) \models \phi$.*

Proof. The proof is, as in the previous cases, by induction on the syntactic structure of ϕ. Assume first $ap \in \Theta$, where p is an ordinary propositional symbol. We need to prove $\mathcal{M}^\Theta, u_\Theta(a) \models p$. By Urfather Closure (Lemma 6.14) we have $u_\Theta(a)p \in \Theta$ and thus by definition of V^Θ we get $u_\Theta(a) \in V^\Theta(p)$, proving $\mathcal{M}, u_\Theta(a) \models p$. Assume now $a\neg p \in \Theta$. Then $u_\Theta(a)\neg p \in \Theta$, and since Θ is open, $u_\Theta(a)p \notin \Theta$. This implies $u_\Theta(a) \notin V^\Theta(p)$, and thus $\mathcal{M}^\Theta, u_\Theta(a) \models \neg p$. Assume $ab \in \Theta$, where b is a nominal. Then b is a right nominal. Thus, by Lemma 6.13, $u_\Theta(b) = \min([b]_\Theta)$. Since $ab \in \Theta$ we also have $u_\Theta(a) = \min([b]_\Theta)$, and thus $u_\Theta(b) = u_\Theta(a)$. This implies $V^\Theta(b) = \{u_\Theta(a)\}$ and thus $\mathcal{M}^\Theta, u_\Theta(a) \models b$, as needed. Now assume $a\neg b \in \Theta$, where b is a nominal. Closure

under the $[\neg]$ rule then gives $bb \in \Theta$. Thus b is a right nominal, and $u_\Theta(b) = \min([b]_\Theta)$, by Lemma 6.13. Therefore $u_\Theta(b) \sim_\Theta b$, and by definition of $\sim_\Theta$ this implies $u_\Theta(b)b \in \Theta$. Since $a\neg b \in \Theta$, Urfather Closure gives $u_\Theta(a)\neg b \in \Theta$. We now have that Θ contains both $u_\Theta(b)b$ and $u_\Theta(a)\neg b$. Since Θ is open this implies $u_\Theta(b) \neq u_\Theta(a)$. By definition of V^Θ we have $V^\Theta(b) = \{u_\Theta(b)\}$, and we thus get $V^\Theta(b) \neq \{u_\Theta(a)\}$. This immediately implies $\mathcal{M}^\Theta, u_\Theta(a) \models \neg b$. Assume now $a\Diamond_i(\phi_1, \ldots, \phi_{\rho(i)}) \in \Theta$, where the formula is not an accessibility formula. Then by Urfather Closure, we have $u_\Theta(a)\Diamond_i(\phi_1, \ldots, \phi_{\rho(i)}) \in \Theta$. Closure under the $[\Diamond]$ rule now gives that there exists nominals $a_1, \ldots, a_{\rho(i)}$ such that

$$u_\Theta(a)\Diamond_i(a_1, \ldots, a_{\rho(i)}) \in \Theta \tag{5}$$

$$a_1\phi_1, a_2\phi_2, \ldots, a_{\rho(i)}\phi_{\rho(i)} \in \Theta. \tag{6}$$

Using (5) and the definition of R_i^Θ gives us

$$(u_\Theta(a), u_\Theta(a_1), \ldots, u_\Theta(a_{\rho(i)})) \in R_i^\Theta. \tag{7}$$

Using (6) and the induction hypothesis gives us

$$\begin{array}{rl} \mathcal{M}^\Theta, u_\Theta(a_1) & \models \phi_1 \\ \mathcal{M}^\Theta, u_\Theta(a_2) & \models \phi_2 \\ & \vdots \\ \mathcal{M}^\Theta, u_\Theta(a_{\rho(i)}) & \models \phi_{\rho(i)}. \end{array} \tag{8}$$

Combining (7) and (8) then immediately gives

$$\mathcal{M}^\Theta, u_\Theta(a) \models \Diamond_i(\phi_1, \ldots, \phi_{\rho(i)}),$$

as required. Now assume $a\neg\Diamond_i(\phi_1, \ldots, \phi_{\rho(i)}) \in \Theta$. Then $u_\Theta(a)\neg\Diamond_i(\phi_1, \ldots, \phi_{\rho(i)}) \in \Theta$ by Urfather Closure. We need to prove $\mathcal{M}^\Theta, u_\Theta(a) \models \neg\Diamond_i(\phi_1, \ldots, \phi_{\rho(i)})$. If there are no nominals $a_1, \ldots, a_{\rho(i)}$ such that $(u_\Theta(a), a_1, \ldots, a_{\rho(i)}) \in R_i^\Theta$, then this holds trivially. Otherwise, let such $a_1, \ldots, a_{\rho(i)}$ be chosen arbitrarily. By definition of R_i^Θ there must be prefixes $b_1, \ldots, b_{\rho(i)}$ such that $a_j = u_\Theta(b_j)$ for all $j = 1, \ldots, \rho(i)$ and such that Θ contains $u_\Theta(a)\Diamond_i(b_1, \ldots, b_{\rho(i)})$. Closure under $[\neg\Diamond]$ now gives that Θ must contain $b_j\neg\phi_j$ for some $j \in \{1, \ldots, \rho(i)\}$. Induction hypothesis implies $\mathcal{M}^\Theta, u_\Theta(b_j) \models \neg\phi_j$, and since $u_\Theta(b_j) = a_j$ this gives $\mathcal{M}^\Theta, a_j \models \neg\phi_j$. Since the nominals $a_1, \ldots, a_{\rho(i)}$ were chosen arbitrarily with the property $(u_\Theta(a), a_1, \ldots, a_{\rho(i)}) \in R_i^\Theta$, we can now conclude $\mathcal{M}^\Theta, u_\Theta(a) \models \neg\Diamond_i(\phi_1, \ldots, \phi_{\rho(i)})$, as required. Assume now $ab\psi \in \Theta$. We need to prove $\mathcal{M}^\Theta, u_\Theta(a) \models b\psi$, that is, $\mathcal{M}^\Theta, u_\Theta(b) \models \psi$. However, this is trivial, as an application of $[@]$ to $ab\psi$ gives $b\psi$ and from this the induction hypothesis implies $\mathcal{M}^\Theta, u_\Theta(b) \models \psi$. Finally, assume $a\neg b\psi \in \Theta$. Then closure under $[\neg@]$ gives $b\neg\psi \in \Theta$. By induction hypothesis this implies $\mathcal{M}^\Theta, u_\Theta(b) \models \neg\psi$. Since $V^\Theta(b) = \{u_\Theta(b)\}$ we can conclude $\mathcal{M}^\Theta, u_\Theta(a) \models b\neg\psi$, and thus $\mathcal{M}^\Theta, u_\Theta(a) \models \neg b\psi$, as required. □

7. Conclusion

In this paper we have systematically considered termination in hybrid tableaus for both prefixed and internalised systems. Our tableau calculi both generalise and simplify earlier work on hybrid tableau systems, and our termination and completeness proofs make the impact of each component of the hybrid language clear. To close this paper, however, we would like to draw attention to some issues that require further attention.

First, an obvious extension of the results above would be to try to add various pure axioms to the calculus of L and see if we could still retain termination and completeness. Now, the urfather method can not be used for extensions with pure axioms in general. The urfather method always produces a finite model, but with pure extensions we can easily make logics that do not have the finite model property. For example, suppose we added a pure axiom for transitivity ($i\Diamond j \wedge j\Diamond k \rightarrow i\Diamond k$) and one for irreflexivity ($i\neg\Diamond i$). Then the formula $A\Diamond p$ introduced in Example 5.1 is satisfiable in the extended logic, but any model for it will be infinite. We hope to invent a more complex urfather method which will work with certain *restricted* classes of pure axioms and logics without the finite model property, but this lies beyond the scope of the present paper.

We'd also like to point out another interesting extension that our approach does not seem to cover, namely hybrid logic enriched with the *difference operator* (see [6]). The difference modality is a modality D such that $M, w \models D\phi$ iff there is some world $v \neq w$ such that $M, v \models \phi$. It is a powerful modality: it can define the global modality ($E\phi$ is just $\phi \vee D\phi$) but the global modality cannot define D. Now, it is easy to devise tableau rules for D. Consider, for example, the following rules (which we give in prefix notation):

$$\frac{\sigma D\phi}{\begin{array}{c}\sigma a\\ \tau\neg a\\ \tau\phi\end{array}}\,[D] \qquad \frac{\sigma\neg D\phi \quad \sigma a \quad \tau\neg a}{\tau\neg\phi}\,[\neg D]$$

In the $[D]$ rule, τ and a are new to the tableau. But if we add these rules to the (prefixed) tableau system, then we don't believe that the resulting system is complete. Here's the intuition. Consider a formula of the form $\sigma(\neg D\phi \wedge bb)$. A saturated tableau with this formula as root will look like this

$$\begin{array}{c}\sigma(\neg D\phi \wedge bb)\\ \big|\ [\wedge]\text{ rule}\\ \sigma\neg D\phi\\ \sigma bb\\ \big|\ [@]\text{ rule}\\ \tau b\end{array}$$

We can not apply the $[\neg D]$ rule to obtain $\tau\neg\phi$ as the tableau branch doesn't contain σc and $\tau\neg c$ for any nominal c. This shows that the present calculus containing $[D]$ and $[\neg D]$ cannot be complete, since the model constructed from the open, saturated branch above will contain the two distinct worlds σ and τ where $\neg D\phi$ will be true in σ but $\neg\phi$ won't be true in τ. But maybe there is a simple fix. Can't we just add a rule which allows us to add formulas of the form σc and $\tau\neg c$ to an open, saturated branch in which there is no nominal that both σ and τ make true? Let us see what would happen. In that case we could extend the tableau above in the following way

$$\begin{array}{c}\sigma(\neg D\phi \wedge bb)\\ \big|\ [\wedge]\text{ rule}\\ \sigma\neg D\phi\\ \sigma bb\\ \big|\ [@]\text{ rule}\\ \tau b\\ \big|\ \text{applying the new rule}\\ \sigma c\\ \tau\neg c\\ \big|\ [\neg D]\text{ rule on }\sigma\neg D\phi, \sigma c, \tau\neg c\\ \tau\neg\phi\end{array}$$

That looks good, but what if ϕ is the formula b? Then the leaf of the tableau above is the formula $\tau\neg b$, and thus the tableau closes! So now it's soundness that fails. Either way, we apparently cannot get both soundness and completeness.

What's going wrong? Well, the problem is that during the course of constructing a tableau branch, we will obtain more and more pairs of formulas of the form $\sigma c, \tau c$, expressing that the prefixes σ and τ must denote the same world. Thus, during the construction of a tableau branch, the number of worlds that get identified must be monotonically increasing. Thus in general there is no point during this construction where we can make sure that two prefixes σ and τ aren't going to denote the same world, and therefore, in general we can't apply $[\neg D]$ in a way which is guaranteed to be sound. In short: since worlds are gradually being identified during tableau

constructions, there seems to be no simple way to extend the tableau systems considered here to cover the difference operator.

Acknowledgements

Thanks to Torben Braüner for valuable comments and suggestions. Also thanks to two anonymous referees for detailed comments. The work of Thomas Bolander is 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] Carlos Areces, Patrick Blackburn, and Maarten Marx. A road-map on complexity for hybrid logics. In *CSL*, pages 307–321, 1999.
[3] P. Blackburn. Internalizing labelled deduction. *Journal of Logic and Computation*, 10:137–168, 2000.
[4] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 2001.
[5] Thomas Bolander and Torben Braüner. Tableau-based decision procedures for hybrid logic. *Journal of Logic and Computation*, 16:737–763, 2006.
[6] Maarten de Rijke. The modal logic of inequality. *Journal of Symbolic Logic*, 57(2):566–584, 1992.
[7] J. Halpern, B. Moskowski, and Z. Manna. A hardware semantics based on temporal intervals. In *ICALP'83*, volume 154 of *Lecture Notes in Computer Science*, pages 278–291. Springer-Verlag, 1983.
[8] M. Tzakova. Tableau calculi for hybrid logics. *Lecture Notes in Computer Science*, 1617:278–292, 1999.
[9] Yde Venema. Derivation rules as anti-axioms in modal logic. *Journal of Symbolic Logic*, 58(3):1003–1034, 1993.
[10] Chaochen Zhou and Michael R. Hansen. *Duration Calculus: A Formal Approach to Real-Time Systems*. Monographs in Theoretical Computer Science. Springer, 2004.