Terminating Tableau Calculi for Hybrid Logics extending K

Thomas Bolander

Technical University of Denmark
Copenhagen, Denmark

Patrick Blackburn

INRIA Grand-Est
Nancy, France

Abstract

This article builds on work by Bolander and Blackburn [7] on terminating tableau systems for the minimal hybrid logic **K**. We provide (for the basic uni-modal hybrid language) terminating tableau systems for a number of non-transitive hybrid logics extending **K**, such as the logic of irreflexive frames, antisymmetric frames, and so on; these systems don't employ loop-checks. We also provide (for hybrid tense logic enriched with the universal modality) a terminating tableau calculus for the logic of transitive frames; this system makes use of loop-checks.

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

1 Introduction

Hybrid logicians like to claim that hybrid logic has two proof-theoretical advantages over orthodox modal logic. The first is that a simple and general completeness result can be proved: any pure axiom is deductively complete with respect to the class of frames it defines. The second is that hybrid logic is an ideal setting for a wide range of proof styles: sequent calculi, tableau systems, resolution, and natural deduction can all be handled straightforwardly. Moreover, the two advantages are additive: when pure formulas are used as extra axioms in (say) a tableau or natural deduction system for the minimal hybrid logic **K**, extended completeness with respect to the frames defined by the pure formulas is typically automatic.

But though these advantages are real, at present it is unclear what relevance (if any) they have for *computational* logic. While the basic hybrid logic **K** is decidable, adding pure axioms can easily yield undecidable logics. Moreover, even if an extension of **K** is known to be decidable, adding the relevant pure axioms to a terminating proof method for **K** will often result in a non-terminating system.

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

The purpose of the present paper is to find terminating tableau methods for hybrid logics richer than **K**. This turns out to be quite tricky to do. Actually, that should come as no surprise. Tableau methods for hybrid logics have been around for almost a decade (Tzakova [12] is the pioneering paper) but the literature is full of examples of non-terminating tableau systems even for the basic hybrid logic **K**. Indeed, it was not until the recent publication of [7] that a terminating tableau system for hybrid **K**, that did not resort to loop-checks or extra side conditions on rules, was presented (in particular [8], the predecessor of [7], employs loop-checks to ensure termination).

The present paper is a direct successor to [7]. Now that termination for **K** has been established (without appealing to loop checks) it is time to look for terminating tableau systems for richer hybrid logics. For most of this paper we will work in basic (uni-modal) hybrid logic, and prove termination results that cover a number of (non-transitive) hybrid definable frame classes (irreflexive frames, antisymmetric frames, intransitive frames, and so on) together with the modally definable class of reflexive frames. In the last section of the article we move to a stronger language (full hybrid tense logic enriched with the universal modality) and prove a termination result that covers transitive frames.

However, as in our previous work, we are also interested in mapping out the techniques required to guarantee termination for the various logics, and in using the weakest methods possible. Thus Section 5 is devoted to finding out which logics have simply terminating tableau systems (that is, systems for which termination can be established without extra side conditions on rules or loop-checks). Section 6 is devoted to finding logics where extra side conditions (without loop-checks) suffice to guarantee termination. Only in Section 7, when we deal with transitivity, do we resort to loop-checks. Along the way we provide a number of counterexamples to the various types of termination. In our view, such counterexamples are almost as important as the termination results, for they help pinpoint the major shifts involved when computing with richer logics. Finally, probably the most vivid lesson we learned form writing this paper is how tricky it is to obtain general termination results, and how much remains to be done; we conclude the paper with a brief discussion of the issues involved.

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 of ordinary propositional symbols and a countably infinite set 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 $a, b, c, d, \ldots$, 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 $a\phi$ asserts 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 is usually written as $@_a\phi$ or $a : \phi$, but we find the lighter notation $a\phi$ more natural for proof-theoretical purposes.

For most of the paper we will work with a modal language containing only a single unary (diamond) modal operator F, whose dual (box) form is G. This language will be called L. It is defined by the following grammar:

$$\phi \ ::= p \mid a \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid a\phi \ \mid F\phi \mid G\phi \qquad (L)$$

Here p is an ordinary propositional symbol and a is a nominal. However in Section 7 we will add the diamond and box forms of the standard *Priorean converse operators* (namely P and H) together with the diamond and box forms of the *universal modality* (namely E and A). So in Section 7 we will be working with what is sometimes called *nominal tense logic* enriched with the universal modality [2]. Now for the semantics. A *frame* for L is a tuple (W, R) where W is a non-empty set (the set of *worlds*) and R is a binary relation on W called the *accessibility relation*. A *model* for L is a tuple (W, R, V) where (W, R) is a frame, and V is a *valuation*: for each proposition symbol or nominal s, $V(s)$ is a subset of W. If s is a nominal then $V(s)$ must be a singleton set. The satisfaction relation $\mathcal{M}, w \models \phi$ is defined inductively, where $\mathcal{M} = (W, R, V)$ is a model, w is an element of W, and ϕ is a formula of L.

$$\begin{array}{ll}
\mathcal{M}, w \models s & \text{iff } w \in V(s), \text{ where } s \text{ is a propositional symbol or a nominal} \\
\mathcal{M}, w \models \neg\phi & \text{iff not } \mathcal{M}, w \models \phi \\
\mathcal{M}, w \models \phi \wedge \psi & \text{iff } \mathcal{M}, w \models \phi \text{ and } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models a\phi & \text{iff } \mathcal{M}, v \models \phi, \text{ where } V(a) = \{v\} \\
\mathcal{M}, w \models F\phi & \text{iff for some } v \in W, (w, v) \in R \text{ and } \mathcal{M}, v \models \phi \\
\mathcal{M}, w \models G\phi & \text{iff for all } v \in W \text{ with } (w, v) \in R, \mathcal{M}, v \models \phi
\end{array}$$

The four additional operators we employ in Section 7 have the following semantics:

$$\begin{array}{ll}
\mathcal{M}, w \models P\phi & \text{iff for some } v \in W, (v, w) \in R \text{ and } \mathcal{M}, v \models \phi \\
\mathcal{M}, w \models H\phi & \text{iff for all } v \in W \text{ with } (v, w) \in R, \mathcal{M}, v \models \phi \\
\mathcal{M}, w \models E\phi & \text{iff for some } v \in W, \mathcal{M}, v \models \phi \\
\mathcal{M}, w \models A\phi & \text{iff for all } v \in W, \mathcal{M}, v \models \phi
\end{array}$$

By convention $\mathcal{M} \models \phi$ means $\mathcal{M}, w \models \phi$ for every element w of W, that is, ϕ is *globally satisfied* in $\mathcal{M}$. A formula ϕ is *valid on* a frame $\mathcal{F} = (\mathcal{W}_{\mathcal{F}}, \mathcal{R}_{\mathcal{F}})$ if and only if ϕ is globally satisfied in all models of the form $(W_{\mathcal{F}}, R_{\mathcal{F}}, V)$, and in such a case we write $\mathcal{F} \models \phi$.

3 An internalised tableau calculus

We will now present an internalised tableau calculus for the hybrid language L. The basic notions for tableaus are defined as usual (see, for example, [7]; this paper also compares in detail internalised and prefixed tableau systems for hybrid logic). The rules of our tableau calculus are given in Figure 1. A tableau branch in this calculus is said to be *closed* if it contains both $a\phi$ and $a\neg\phi$ for some nominal a and formula ϕ. A tableau branch which is not closed is called *open*. A tableau

$$\frac{a\neg a}{\perp}\ (\neg_1)^1 \qquad\qquad \frac{a\neg b,\, ba}{\perp}\ (\neg_2)^1$$

$$\frac{a(\phi \wedge \psi)}{\begin{array}{c} a\phi \\ a\psi \end{array}}\ (\wedge) \qquad\qquad \frac{a(\phi \vee \psi)}{a\phi \mid a\psi}\ (\vee)$$

$$\frac{aF\phi}{\begin{array}{c} aFb \\ b\phi \end{array}}\ (F)^2 \qquad\qquad \frac{aG\phi,\, aFb}{b\phi}\ (G)$$

$$\frac{ab\phi}{b\phi}\ (@)$$

$$\frac{bc,\, ab}{ac}\ (Nom) \qquad\qquad \frac{a\phi,\, ab}{b\phi}\ (Id)^3$$

[1] $\perp$ denotes a closing tableau.
[2] If ϕ is a nominal then $aF\phi$ is a root subformula. The nominal b is new to the tableau.
[3] If $a\phi$ is of the form aFc then it is a root subformula.

Fig. 1. Tableau calculus for the hybrid language L.

branch is said to be *saturated* if no further rules apply to it. The first formula on a tableau branch is called the *root formula* of the branch. We denote the root formula of a tableau branch Θ by $root_\Theta$. The nominals occurring in $root_\Theta$ are called *root nominals* of Θ. Other nominals are called *non-root nominals*. A formula $a\phi$ is said to be a *quasi-subformula* of a formula $b\psi$ if ϕ is a subformula of ψ. A formula $a\phi$ occurring on a tableau branch Θ is called a *root subformula* on Θ if it is a quasi-subformula of $root_\Theta$. A formula of the form aFb on Θ is called an *accessibility formula* if it is the first conclusion of an application of rule (F). The intended interpretation of an accessibility formula aFb is that b denotes a world accessible from a. In each of the rules of Figure 1, the leftmost premise is called the *principal premise*. If $a\phi$ and $b\psi$ are formulas on a tableau branch Θ, then $b\psi$ is said to be *produced* by $a\phi$ if $b\psi$ is one of the conclusions of a rule application with principal premise $a\phi$. The formula $b\psi$ is said to be *indirectly produced* by $a\phi$ if there exists a sequence of formulas $a\phi, a_1\phi_1, a_2\phi_2, \ldots, a_n\phi_n, b\psi$ in which each formula is produced by its predecessor. All formulas occurring in the tableau rules are expressed in *negation normal form* (NNF), that is, all negation symbols are immediately in front of nominals or propositional symbols (see [11] for the procedure involved). Thus when constructing a tableau in this calculus, the root formula first needs to be put in negation normal form. Note that the rule (F) is the only rule that can introduce new nominals to a tableau branch. We impose two general constraints on the construction of tableaus:

- The rule (F) is never applied twice to the same premise on the same branch.

(ref)	aFa	reflexivity	$\forall x(x < x)$
(irr)	$aG\neg a$	irreflexivity	$\forall x \neg (x < x)$
(sym)	$aGFa$	symmetry	$\forall x, y(x < y \rightarrow y < x)$
(asym)	$aGG\neg a$	asymmetry	$\forall x, y(x < y \rightarrow \neg y < x)$
(antisym)	$aG(a \vee G\neg a)$	antisymmetry	$\forall x, y(x < y \wedge y < x \rightarrow x = y)$
(trans)	$a(GG\neg b \vee Fb)$	transitivity	$\forall x, y, z(x < y \wedge y < z \rightarrow x < z)$
(intrans)	$a(GG\neg b \vee G\neg b)$	intransitivity	$\forall x, y, z(x < y \wedge y < z \rightarrow \neg x < z)$
(trich)	$a(b \vee Fb \vee bFa)$	trichotomy	$\forall x, y(x < y \vee y < x \vee x = y)$
(univ)	aFb	universality	$\forall x, y(x < y)$
(serial)	$aF\top$	seriality	$\forall x \exists y(x < y)$
(uniq)	$a(G\neg b \vee Gb)$	uniqueness	$\forall x, y, z(x < y \wedge x < z \rightarrow y = z)$
(tree)	$a(bG\neg a \vee cG\neg a \vee bc)$	tree-like	$\forall x, y, z(x < z \wedge y < z \rightarrow x = y)$
(euc)	$a(G\neg b \vee G\neg c \vee bFc)$	euclidean	$\forall x, y, z(x < y \wedge x < z \rightarrow x < z)$
(dense)	$a(G\neg b \vee FFb)$	density	$\forall x, y(x < y \rightarrow \exists z(x < z \wedge z < y))$

Fig. 2. A collection of pure formulas in L and their defining frame properties.

- A formula is never added to a tableau branch where it already occurs.

The tableau calculus for L consisting of all the rules of Figure 1 is called $\mathtt{L}$. To express that a formula $a\phi$ occurs on a tableau branch Θ we often simply write $a\phi \in \Theta$. In this case we sometimes say that ϕ is *true at a* on Θ.

4 Adding axioms

A formula of L is called a *pure formula* if it doesn't contain any ordinary propositional symbols. Pure formulas can be used to define frame properties of hybrid logics. A formula ϕ is said to *define* a class of frames F if and only if: $\mathcal{F} \models \phi \Leftrightarrow \mathcal{F} \in \mathsf{F}$. That is, a formula defines the class of frames it is valid on. When we say a hybrid formula defines a certain property (for example, transitivity) we mean it defines the class of all frames with that property. Figure 2 gives a list of some pure formulas and the frame properties defined by these formulas. It is well-known that the properties irreflexivity, asymmetry, antisymmetry, intransitivity, trichotomy and universality are *not* definable in ordinary modal logic (see [2]). Note that all pure formulas of Figure 2 are expressed as satisfaction statements in negation normal form. In the following we will assume pure formulas are always expressed in this form. This allows us to use pure formulas as axioms in our tableau calculi. Let $\mathtt{H}$ be a tableau calculus, and let $\mathtt{Axiom}$ be a set of pure formulas. We let $\mathtt{H} + \mathtt{Axiom}$ denote the tableau system that results from $\mathtt{H}$ by using the formulas of $\mathtt{Axiom}$ as axioms. That is, on any tableau branch Θ of $\mathtt{H}$ we are free to add formulas on the form $(a\phi)[b/a, b_1/a_1, \ldots, b_n/a_n]$, where $a\phi$ is a formula in $\mathtt{Axiom}$, $a, a_1, \ldots, a_n$ are the nominals occurring in $a\phi$, and $b, b_1, \ldots, b_n$ are nominals already occurring on Θ. Let Θ be a tableau branch in the calculus $\mathtt{H}+\mathtt{Axiom}$. A formula $b\psi \in \Theta$ is called an *axiom subformula* on Θ if it is a quasi-subformula of a formula $a\phi \in \mathtt{Axiom}$, modulo a renaming of the nominals in ϕ. In [3], Blackburn gives a tableau calculus $\mathtt{H}$ for a hybrid logic similar to $\mathtt{L}$. He proves that for any set of pure formulas $\mathtt{Axiom}$, the calculus $\mathtt{H}+\mathtt{Axiom}$ is sound and complete with respect to the class of frames defined by $\mathtt{Axiom}$. However, there are no general results stating whether the calculus $\mathtt{H}+\mathtt{Axiom}$ will be terminating or not. In the following sections we will try to investigate which pure formulas can be added as axioms while retaining both termination and completeness with respect to the class of frames defined by the axioms.

5 Simple termination

In this section we will investigate which pure formulas can be added as axioms without needing to ensure termination by imposing extra side-conditions on the rules or by resorting to loop-checks. As we shall see, simple termination can be proved for combinations of (ref), (irr), (asym), and (intrans). Most of the lemmas proved in the course of the discussion will be re-used in the following section. We begin by noting that the following lemma is easily proved by checking the effect of the rules of $\mathtt{L}$.

Lemma 5.1 (Subformula Property) *Let $\mathtt{Axiom}$ be a set of pure formulas in L, and let Θ be a tableau branch of $\mathtt{L}+\mathtt{Axiom}$. Any formula $a\phi$ occurring on Θ is either a root subformula, an axiom subformula or an accessibility formula.*

Let Θ be a tableau branch in any calculus. If a nominal b has been introduced to the branch by applying (F) to a premise $a\phi$ then we say b is *generated* by a on Θ, and we write $a \prec_\Theta b$. We use $\prec_\Theta^*$ to denote the reflexive and transitive closure of $\prec_\Theta$.

A pure L-formula is called *non-existential* if in all subformulas of the form $F\phi$, ϕ is a nominal. We choose the term 'non-existential' to refer to these formulas, since if an axiom is non-existential then it cannot introduce new nominals into a tableau: only the rule (F) can introduce new nominals, but it doesn't apply to axiom subformulas of the form $aF\phi$ where ϕ is a nominal. All pure formulas in Figure 2 except (serial) and (dense) are non-existential. Let Θ be a tableau branch in any calculus. The set of nominals occurring on Θ is denoted Nom_Θ.

Lemma 5.2 *Let $\mathtt{Axiom}$ be a set of non-existential formulas in L, and let Θ be a tableau branch in $\mathtt{L}+\mathtt{Axiom}$. The graph $G = (Nom_\Theta, \prec_\Theta)$ is a finite set of wellfounded, finitely branching trees.*

Proof. That G is wellfounded follows from the observation that if $a \prec_\Theta b$, then the first occurrence of a on Θ is before the first occurrence of b. That the graph is a finite set of trees follows from the fact that each nominal in Nom_Θ can be generated by at most one other nominal, and that each nominal in Nom_Θ must have one of the finitely many root nominals as an ancestor. We will now show that G is finitely branching. Given a nominal a, we need to show that there can only be finitely many distinct nominals b satisfying $a \prec_\Theta b$. Each nominal b satisfying $a \prec_\Theta b$ is by definition generated by applying (F) to a premise of the form $aF\phi$, where either ϕ is not a nominal or $aF\phi$ is a root subformula. Since only non-existential axioms have been introduced on Θ, all formulas of the form $aF\phi$ where ϕ is not a nominal must be root subformulas, according to Lemma 5.1. Since there can only be finitely many root subformulas of the form $aF\phi$ for any nominal a, only finitely many new nominals can have been generated from a. This shows that G is finitely branching. □

Lemma 5.3 *Let* `Axiom` *be a set of non-existential formulas in L. A tableau branch Θ in* $\mathtt{L} + \mathtt{Axiom}$ *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 'if' direction is trivial. To prove the 'only if' direction, let Θ be any infinite tableau branch. Note that according to our tableau conventions all formulas occurring on the infinite branch Θ are distinct. We will first prove that Nom_Θ is infinite. Assume to obtain a contradiction that it is finite. According to Lemma 5.1, all formulas on Θ are either root subformulas, axiom subformulas or accessibility formulas. Since Nom_Θ is finite, there can only be finitely many distinct root subformulas and accessibility formulas on Θ. Since `Axiom` is finite, there can only be finitely many distinct axiom subformulas on Θ. Thus, the number of distinct formulas on Θ must be finite, contradicting our assumption. Thus we have proven Nom_Θ to be infinite. According to Lemma 5.2, the graph $G = (Nom_\Theta, \prec_\Theta)$ is a finite set of wellfounded, finitely branching trees. Since G has now been shown to be infinite, it must—by König's Lemma—contain an infinite path. An infinite path in G is an infinite chain of prefixes $a_1 \prec_\Theta a_2 \prec_\Theta a_3 \prec_\Theta \cdots$. □

An occurrence of a nominal in a formula is called *negative* if the nominal is immediately preceded by a negation symbol. A pure formula in negation normal form is called *negative* if all nominal occurrences are negative.

Lemma 5.4 *Let* `Axiom` *denote a set of negative, non-existential formulas, and let Θ be a tableau branch in* $\mathtt{L} + \mathtt{Axiom}$. *If $ab \in \Theta$ for a pair of nominals a, b then b is a root nominal.*

Proof. Assume $ab \in \Theta$. Lemma 5.1 implies that ab is either a root subformula or an axiom subformula. Assume first ab is an axiom subformula. Then by assumption on the set of axioms, ab have been indirectly produced by an axiom subformula of the form cFd or $c\neg d$. However, as no rule of `L` accepts an axiom subformula of the form cFd as principal premise, ab can not have been indirectly produced by cFd. Assume instead ab was indirectly produced by $c\neg d$. The only rules that possibly apply to $c\neg d$ are (Id), $(\neg_1)$ and $(\neg_2)$. However, if (Id) is applied to a principal premise of the form $c\neg d$ it produces a conclusion of the same form; and the rules $(\neg_1)$ and $(\neg_2)$ produce no conclusions. Thus ab can not have been indirectly produced by $c\neg d$ either. This means ab can not be an axiom subformula. It must therefore be a root subformula, and we immediately get that b is a root nominal. □

A pure formula is called F-free if it doesn't contain any occurrences of the F-operator. Every F-free formula is obviously non-existential. The F-free formulas of

Figure 2 are (irr), (asym), (antisym), (intrans), (uniq) and (tree).

Lemma 5.5 *Let* $\mathtt{Axiom}$ *be a set of* F*-free formulas plus possibly the axiom* (ref). *Let* Θ *a tableau branch in* $\mathtt{L} + \mathtt{Axiom}$*. If* $aFb \in \Theta$ *and* b *is a non-root nominal then* $a \prec_\Theta b$ *or* $a = b$*.*

Proof. Suppose $aFb \in \Theta$ where b is non-root. We need to prove that $a \prec_\Theta b$ or $a = b$. First note that aFb can not be a root subformula as b is assumed to be non-root. According to Lemma 5.1, aFb is thus either an accessibility formula or an axiom subformula. If it is an accessibility formula it is the first conclusion of an application of (F), and thus $a \prec_\Theta b$, by definition. If it is an axiom subformula, it is indirectly produced by an instance of one of the axioms. By assumption on $\mathtt{Axiom}$, the only axioms that can contain subformulas of the form Fc for a nominal c are instances of (ref). Thus aFb must be indirectly produced by an instance of this axiom. As none of the rules of $\mathtt{L}$ accepts an axiom subformula of the form aFb as a principal premise, aFb must itself be an instance of (ref). Hence $a = b$, as required. $\square$

Let Θ be a tableau branch in any calculus, and let a be a nominal occurring on Θ. We define $m_\Theta(a)$ by

$$m_\Theta(a) = \max\{|a\phi| : a\phi \in \Theta \text{ and } a\phi \text{ is a root subformula}\},$$

where $|a\phi|$ is the length of the formula $a\phi$. If there are no root subformulas $a\phi$ on Θ we let $m_\Theta(a) = -\infty$. Let Θ be a branch of a tableau, and let a be a nominal occurring on Θ. The *depth* of a wrt. Θ, denoted $d_\Theta(a)$, is the length of the unique path in $(Nom_\Theta, \prec_\Theta)$ connecting a root nominal with a. The uniqueness of such a path is guaranteed by Lemma 5.2.

Lemma 5.6 (Decreasing length) *Let* $\mathtt{Axiom}$ *be a finite set of negative,* F*-free formulas plus possibly the axiom* (ref)*. Let* Θ *be a tableau branch in* $\mathtt{L} + \mathtt{Axiom}$*. For any nominal* a *on* Θ*,* $m_\Theta(a) \leq |root_\Theta| - d_\Theta(a)$*.*

Proof. The proof is by induction on the depth of the nominal a. The base case where a has depth 0 is trivial, as $m_\Theta(a) \leq |root_\Theta|$ is satisfied for any nominal a. For the induction step let b be a nominal of depth > 0, that is, a non-root nominal. We need to prove $m_\Theta(b) \leq |root_\Theta| - d_\Theta(b)$ under the assumption that the inequality holds for all nominals of lower depth. If there are no root subformulas true at b on Θ then $m_\Theta(b) = -\infty$ and there is nothing to prove. Otherwise, let ϕ be a formula of maximal length for which $b\phi$ is a root subformula on Θ. Then $b\phi$ has been introduced on Θ by applying one of the rules of $\mathtt{L}$ to a root subformula. The formula $b\phi$ can not have been introduced on Θ by applying either $(\wedge)$ or $(\vee)$, since this contradicts the maximality of ϕ. We can furthermore assume that $b\phi$ has not been introduced by an application of (Nom), since in this case $|b\phi| = 2$, and there must be another formula $b\psi$ of the same length that has been introduced by an application of (F) (the application of (F) generating the nominal b). Assume $b\phi$ has been introduced by applying (@) to some root subformula $cb\phi$. Then b must be a root nominal, contradicting our assumption. Thus (@) can not have been the rule producing $b\phi$. Now assume $b\phi$ has been introduced by applying (Id) to premises $a\phi$

and ab. Then according to Lemma 5.4, b must be a root nominal, which is again a contradiction. Thus $b\phi$ can not have been produced by (Id) either. Therefore $b\phi$ has been introduced by one of the rules (F) or (G). In the case of (F), $b\phi$ must have been introduced together with a formula of the form aFb from a premise $aF\phi$. By definition of $\prec_\Theta$ we then have $a \prec_\Theta b$, and thus $d_\Theta(b) = d_\Theta(a) + 1$. The induction hypothesis then gives us $m_\Theta(a) \leq |root_\Theta| - d_\Theta(a)$. Hence we get

$$m_\Theta(b) = |b\phi| = |aF\phi| - 1 \leq m_\Theta(a) - 1 \leq |root_\Theta| - d_\Theta(a) - 1 = |root_\Theta| - d_\Theta(b). \quad (1)$$

This is the required conclusion. Now consider the case where $b\phi$ is introduced by an application of (G). Then $b\phi$ must be introduced by applying (G) to a pair of premises on Θ of the form $aG\phi$, aFb. Since $b\phi$ is assumed to be of maximal length, we must have $a \neq b$. Consider the premise aFb on Θ. The nominal b is non-root, so we can apply Lemma 5.5 to conclude $a \prec_\Theta b$. Thus again we get $d_\Theta(b) = d_\Theta(a) + 1$ and by induction hypothesis, $m_\Theta(a) \leq |root_\Theta| - d_\Theta(a)$. This implies the same series of inequalities as in (1), except $aF\phi$ is replaced by $aG\phi$. □

Lemma 5.7 *Let* `Axiom` *denote a set of non-existential formulas, and let Θ be a tableau branch in the calculus* $\mathtt{L} + \mathtt{Axiom}$. *Assume Θ satisfies $\forall a \in Nom_\Theta(m_\Theta(a) \leq |root_\Theta| - d_\Theta(a))$. Then Θ is finite.*

Proof. Assume to obtain a contradiction that Θ is infinite. Since `Axiom` is a finite set of non-existential formulas, Lemma 5.3 then implies there is an infinite chain $a_0 \prec_\Theta a_1 \prec_\Theta a_2 \prec_\Theta \cdots$. For all i we get $d_\Theta(a_{i+1}) = d_\Theta(a_i) + 1$, and thus, for all i, $d_\Theta(a_i) \geq i$. Using the assumption on Θ this implies

$$\forall i \in \mathbb{N}(m_\Theta(a_i) \leq |root_\Theta| - i). \quad (2)$$

Since $a_i \prec_\Theta a_{i+1}$ for all i, there must for each i exist a formula ϕ_i such that Θ contains $a_i F\phi_i$ and such that a_{i+1} has been introduced to Θ by applying (F) to this formula. By assumption on `Axiom`, all ϕ_i must be root subformulas. Hence for all i we get $m_\Theta(a_i) > 0$. However, this is in direct contradiction to (2). □

Combining Lemma 5.6 with Lemma 5.7 immediately gives us the following termination result.

Lemma 5.8 *Let* `Axiom` *denote a finite set of negative, F-free formulas plus possibly the axiom* (ref). *Any tableau in the calculus* $\mathtt{L} + \mathtt{Axiom}$ *is finite.*

The negative, F-free formulas of Figure 2 are (irr), (asym) and (intrans). The termination theorem above does therefore *not* give us termination for the following axioms: (sym), (antisym), (trans), (trich), (univ), (serial), (uniq), (tree), (euc) and (dense). In fact, if we add just a single of these axioms to the calculus, it will no longer terminate. For the axioms that are not non-existential, (serial) and (dense), this is simply because the axioms contain a subformula of the form $F\phi$ where ϕ is not a nominal, and from such a formula a new nominal can always be generated. For the axioms (sym), (antisym), (trans), (trich), (univ) and (uniq), counter-examples to termination is given in the Figures 3–8. Non-termination of the remaining axioms, (tree) and (euc), is left as a simple exercise for the reader.

$$
\begin{array}{rl}
 & \begin{array}{lll} 1 & a_0FFb & \\ 2 & bGFFb & \end{array} \\
X & \left\{ \begin{array}{lll} 3 & a_0Fa_1 & (F) \text{ on } 1 \\ 4 & a_1Fb & (F) \text{ on } 1 \\ 5 & a_1GFa_1 & (\text{sym}) \\ 6 & bFa_1 & (G) \text{ on } 5{,}4 \\ 7 & a_1FFb & (G) \text{ on } 2{,}6 \end{array} \right. \\
X[a_1/a_0, a_2/a_1] & \left\{ \begin{array}{lll} 8 & a_1Fa_2 & (F) \text{ on } 7 \\ 9 & a_2Fb & (F) \text{ on } 7 \\ 10 & a_2GFa_2 & (\text{sym}) \\ 11 & bFa_2 & (G) \text{ on } 10{,}9 \\ 12 & a_2FFb & (G) \text{ on } 2{,}11 \end{array} \right. \\
 & \begin{array}{ll} \vdots & \vdots \end{array}
\end{array}
$$

Fig. 3. Non-termination with (sym).

$$
\begin{array}{rl}
 & \begin{array}{lll} 1 & a_0FFa_0 & \\ 2 & a_0Fa_1 & (F) \text{ on } 1 \\ 3 & a_1Fa_0 & (F) \text{ on } 1 \end{array} \\
X & \left\{ \begin{array}{lll} 4 & a_1G(a_1 \vee G\neg a_1) & (\text{antisym}) \\ 5 & a_0(a_1 \vee G\neg a_1) & (G) \text{ on } 4{,}3 \\ 6 & a_0a_1 & (\vee) \text{ on } 5 \\ 7 & a_1FFa_0 & (Id) \text{ on } 1{,}6 \\ 8 & a_1Fa_2 & (F) \text{ on } 7 \\ 9 & a_2Fa_0 & (F) \text{ on } 7 \end{array} \right. \\
X[a_2/a_1, a_3/a_2] & \left\{ \begin{array}{lll} 10 & a_2G(a_2 \vee G\neg a_2) & (\text{antisym}) \\ 11 & a_0(a_2 \vee G\neg a_2) & (G) \text{ on } 10{,}9 \\ 12 & a_0a_2 & (\vee) \text{ on } 11 \\ 13 & a_2FFa_0 & (Id) \text{ on } 1{,}12 \\ 14 & a_2Fa_3 & (F) \text{ on } 13 \\ 15 & a_3Fa_0 & (F) \text{ on } 13 \end{array} \right. \\
 & \begin{array}{ll} \vdots & \vdots \end{array}
\end{array}
$$

Fig. 4. Non-termination with (antisym).

$$
\begin{array}{rl}
 & \begin{array}{lll} 1 & a_0(Fp \wedge GFp) & \\ 2 & a_0Fp & (\wedge) \text{ on } 1 \\ 3 & a_0GFp & (\wedge) \text{ on } 1 \\ 4 & a_0Fa_1 & (F) \text{ on } 2 \end{array} \\
X & \left\{ \begin{array}{lll} 5 & a_1Fp & (G) \text{ on } 3{,}4 \\ 6 & a_1Fa_2 & (F) \text{ on } 5 \\ 7 & a_0(GG\neg a_2 \vee Fa_2) & (\text{trans}) \\ 8 & a_0Fa_2 & (\vee) \text{ on } 7 \end{array} \right. \\
X[a_2/a_1, a_3/a_2] & \left\{ \begin{array}{lll} 9 & a_2Fp & (G) \text{ on } 3{,}8 \\ 10 & a_2Fa_3 & (F) \text{ on } 9 \\ 11 & a_0(GG\neg a_3 \vee Fa_3) & (\text{trans}) \\ 12 & a_0Fa_3 & (\vee) \text{ on } 11 \end{array} \right. \\
 & \begin{array}{ll} \vdots & \vdots \end{array}
\end{array}
$$

Fig. 5. Non-termination with (trans).

$$
\begin{array}{rl}
 & \begin{array}{lll} 1 & a_0GFp & \\ 2 & a_0Fa_0 & (\text{univ}) \\ 3 & a_0Fp & (G) \text{ on } 1{,}2 \\ 4 & a_0Fa_1 & (F) \text{ on } 3 \end{array} \\
X & \left\{ \begin{array}{lll} 5 & a_1Fp & (G) \text{ on } 1{,}4 \\ 6 & a_1Fa_2 & (F) \text{ on } 5 \\ 7 & a_0Fa_2 & (\text{univ}) \end{array} \right. \\
X[a_2/a_1, a_3/a_2] & \left\{ \begin{array}{lll} 8 & a_2Fp & (G) \text{ on } 1{,}7 \\ 9 & a_2Fa_3 & (F) \text{ on } 8 \\ 10 & a_0Fa_3 & (\text{univ}) \end{array} \right. \\
 & \begin{array}{ll} \vdots & \vdots \end{array}
\end{array}
$$

Fig. 6. Non-termination with (univ).

5.1 Completeness

In the following we will assume that we have fixed a function σ that to each tableau branch Θ and each non-empty set $N \subseteq Nom_\Theta$ picks out an element of N. The function value $\sigma(\Theta, N)$ will most often be written $\sigma_\Theta(N)$ or simply $\sigma_\Theta N$.

Definition 5.9 (Urfathers) Let Θ be a tableau branch in any calculus, and let a be a nominal occurring on Θ. The *urfather* of a on Θ, denoted $u_\Theta(a)$, is defined by

$$
u_\Theta(a) = \begin{cases} \sigma_\Theta\{b \mid ab \in \Theta\} & \text{if } \{b \mid ab \in \Theta\} \neq \emptyset \\ a & \text{otherwise.} \end{cases}
$$

A nominal a is called an *urfather* on Θ if $a = u_\Theta(b)$ for some nominal b.

Lemma 5.10 *Let Θ be a saturated tableau branch in a calculus containing (Id) and (Nom). Then we have the following properties:*

$$
\begin{array}{llll}
 & 1 & a_0(Fp \wedge GFp) & \\
 & 2 & a_0Fp & (\wedge) \text{ on } 1 \\
 & 3 & a_0GFp & (\wedge) \text{ on } 1 \\
 & 4 & a_0Fa_1 & (F) \text{ on } 2 \\
X \left\{ \begin{array}{l} \\ \\ \\ \\ \end{array} \right. & \begin{array}{l} 5 \\ 6 \\ 7 \\ 8 \end{array} & \begin{array}{l} a_1Fp \\ a_1Fa_2 \\ a_0(a_2 \vee Fa_2 \vee a_2Fa_0) \\ a_0Fa_2 \end{array} & \begin{array}{l} (G) \text{ on } 3,4 \\ (F) \text{ on } 5 \\ \text{(trich)} \\ (\vee) \text{ on } 7 \end{array} \\
X[a_2/a_1, a_3/a_2] \left\{ \begin{array}{l} \\ \\ \\ \\ \end{array} \right. & \begin{array}{l} 9 \\ 10 \\ 11 \\ 12 \end{array} & \begin{array}{l} a_2Fp \\ a_2Fa_3 \\ a_0(a_3 \vee Fa_3 \vee a_3Fa_0) \\ a_0Fa_3 \end{array} & \begin{array}{l} (G) \text{ on } 3,8 \\ (F) \text{ on } 9 \\ \text{(trich)} \\ (\vee) \text{ on } 11 \end{array} \\
 & \vdots & \vdots &
\end{array}
$$

Fig. 7. Non-termination with (trich).

$$
\begin{array}{llll}
 & 1 & a_0FFp & \\
 & 2 & a_0Fa_1 & (F) \text{ on } 1 \\
 & 3 & a_1Fp & (F) \text{ on } 1 \\
X \left\{ \begin{array}{l} \\ \\ \\ \\ \\ \end{array} \right. & \begin{array}{l} 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} & \begin{array}{l} a_1Fa_2 \\ a_0(G\neg a_2 \vee Ga_2) \\ a_0Ga_2 \\ a_1a_2 \\ a_2Fp \end{array} & \begin{array}{l} (F) \text{ on } 3 \\ \text{(uniq)} \\ (\vee) \text{ on } 5 \\ (G) \text{ on } 6,2 \\ (Id)\ 3,7 \end{array} \\
X[a_{i+1}/a_i] \left\{ \begin{array}{l} \\ \\ \\ \\ \\ \end{array} \right. & \begin{array}{l} 9 \\ 10 \\ 11 \\ 12 \\ 13 \end{array} & \begin{array}{l} a_2Fa_3 \\ a_1(G\neg a_3 \vee Ga_3) \\ a_1Ga_3 \\ a_2a_3 \\ a_3Fp \end{array} & \begin{array}{l} (F) \text{ on } 8 \\ \text{(uniq)} \\ (\vee) \text{ on } 10 \\ (G) \text{ on } 11,4 \\ (Id) \text{ on } 8,12 \end{array} \\
 & \vdots & \vdots &
\end{array}
$$

Fig. 8. Non-termination with (uniq).

(i) *If $a\phi$ is a root subformula or an axiom subformula not on the form aFc then $u_\Theta(a)\phi \in \Theta$.*
(ii) *If $ab \in \Theta$ then $u_\Theta(a) = u_\Theta(b)$.*
(iii) *If a is an urfather on Θ then $u_\Theta(a) = a$.*

Proof. First we prove (i). Assume $a\phi$ is a root subformula or an axiom subformula not on the form aFc. If $u_\Theta(a) = a$ then there is nothing to prove. So assume $u_\Theta(a) = \sigma_\Theta\{b \mid ab \in \Theta\}$. Then $au_\Theta(a) \in \Theta$, and by applying (Id) to premises $a\phi$ and $au_\Theta(a)$ we get $u_\Theta(a)\phi$ as needed. We now prove (ii). Assume $ab \in \Theta$. To prove $u_\Theta(a) = u_\Theta(b)$ it suffices to prove that for all nominals c, $ac \in \Theta \Leftrightarrow bc \in \Theta$. So let c be an arbitrary nominal. If $ac \in \Theta$ then we can apply (Id) to premises ac and ab to obtain the conclusion bc, as required. If conversely $bc \in \Theta$ then we can apply (Nom) to premises bc and ab to obtain the conclusion ac, as required. We finally prove (iii). Assume a is an urfather. Then $a = u_\Theta(b)$ for some b. If $b = a$ we are done. Otherwise we have $a = u_\Theta(b) = \sigma_\Theta\{c \mid bc \in \Theta\}$ and thus $ba \in \Theta$. This implies $a = u_\Theta(b) = u_\Theta(a)$, using item (ii). □

Let Θ be an open, saturated tableau branch in any calculus. We can now define a model $\mathcal{M}^\Theta = (W^\Theta, R^\Theta, V^\Theta)$ by

$$
\begin{aligned}
W^\Theta &= \{u_\Theta(a) \mid a \text{ is a nominal occurring on } \Theta\} \\
R^\Theta &= \{(a, u_\Theta(b)) \in W^2 \mid aFb \in \Theta\} \\
V^\Theta(p) &= \{a \in W \mid ap \in \Theta\} \\
V^\Theta(a) &= \{u_\Theta(a)\}.
\end{aligned}
$$

Lemma 5.11 *Let* `Axiom` *denote a finite set of F-free formulas plus possibly the axiom* (ref). *Let Θ be an open, saturated tableau branch in* $\mathtt{L} + \mathtt{Axiom}$. *If a is an urfather and $a\phi$ is a root subformula or axiom subformula occurring on Θ then $\mathcal{M}^\Theta, a \models \phi$.*

Proof. The proof is by induction on the syntactic structure of ϕ, where ϕ is assumed to be on negation normal form. The base cases are $\phi = p$ and $\phi = \neg p$

for propositional symbols p and $\phi = b$ and $\phi = \neg b$ for nominals b. The cases $\phi = p$ and $\phi = \neg p$ are trivial. For the case $\phi = b$ assume $ab \in \Theta$, where a is an urfather and b is a nominal. Then by item (ii) of Lemma 5.10, $u_\Theta(a) = u_\Theta(b)$. Since a is an urfather, item (iii) of Lemma 5.10 implies $u_\Theta(a) = a$. Hence we get $V^\Theta(b) = \{u_\Theta(b)\} = \{u_\Theta(a)\} = \{a\}$, and therefore $\mathcal{M}^\Theta, a \models b$, as needed. Now assume $a\neg b \in \Theta$, where a is an urfather and b is a nominal. By closure under the rule $(\neg_1)$ we must have $a \neq b$. By closure under the rule $(\neg_2)$ we must have $ba \notin \Theta$. This implies $u_\Theta(b) \neq a$. Therefore we get $V^\Theta(b) = \{u_\Theta(b)\} \neq \{a\}$, and hence $\mathcal{M}^\Theta, a \models \neg b$, as required. This concludes the base cases. We now turn to the induction step. The cases where the formula has the form $\psi \vee \chi$ or $\psi \wedge \chi$ are trivial. Assume now $aF\phi \in \Theta$ where a is an urfather. By assumption on `Axiom`, $aF\phi$ can only be an axiom subformula if it is an instance of (ref). If it is an instance of (ref) then $\phi = a$, and $aFa \in \Theta$. Thus we get $(a, u_\Theta(a)) \in R^\Theta$, and since a is an urfather, $u_\Theta(a) = a$. In other words we have $(a, a) \in R^\Theta$ implying $\mathcal{M}^\Theta, a \models Fa$, as required. If $aF\phi$ is a root subformula, closure under (F) implies the existence of a nominal b such that $aFb, b\phi \in \Theta$. Since $aF\phi$ is a root subformula, $b\phi$ is as well, and we can thus apply item (i) of Lemma 5.10 to conclude $u_\Theta(b)\phi \in \Theta$. The induction hypothesis applied to $u_\Theta(b)\phi \in \Theta$ gives $\mathcal{M}^\Theta, u_\Theta(b) \models \phi$. Since $aFb \in \Theta$ we also get $(a, u_\Theta(b)) \in R^\Theta$. Combining these two facts immediately gives us $\mathcal{M}^\Theta, a \models F\phi$, as required. Now assume $aG\phi \in \Theta$ where a is an urfather. If $aG\phi$ is an axiom subformula then ϕ doesn't contain any occurrences of F, by assumption on `Axiom`. We need to prove $\mathcal{M}^\Theta, a \models G\phi$. If there is no nominal b such that $(a, b) \in R^\Theta$, then this holds trivially. Otherwise, let such a b be chosen arbitrarily. We then need to prove $\mathcal{M}^\Theta, b \models \phi$. By definition of R^Θ there must be a nominal b' such that $b = u_\Theta(b')$ and such that $aFb' \in \Theta$. Closure under (G) gives that Θ contains $b'\phi$. Since $b'\phi$ is either a root subformula or an axiom subformula not containing F, we can apply item (i) of Lemma 5.10 to conclude $b\phi \in \Theta$. Thus by induction hypothesis, $\mathcal{M}^\Theta, b \models \phi$, as required. Finally, assume $ab\phi \in \Theta$ where a is an urfather. If $ab\phi$ is an axiom subformula then ϕ doesn't contain F, by assumption on `Axiom`. We need to prove $\mathcal{M}^\Theta, a \models b\phi$. By definition, $V^\Theta(b) = \{u_\Theta(b)\}$, so what we need to prove is $\mathcal{M}^\Theta, u_\Theta(b) \models \phi$. Closure under (@) gives us $b\phi \in \Theta$. Since $b\phi$ is either a root subformula or an axiom subformula not containing F, we can apply item (i) of Lemma 5.10 to conclude $u_\Theta(b)\phi \in \Theta$. From this we immediately get $\mathcal{M}^\Theta, u_\Theta(b) \models \phi$, as required. □

Lemma 5.12 (Gargov, Goranko [9]) *Let S be a non-empty set of nominals and let `Axiom` be a set of pure formulas closed under uniform substitution of nominals in S for nominals. Let $\mathcal{M} = (W, R, V)$ be a model based on a frame $\mathcal{F} = (W, R)$ such that `Axiom` is globally satisfied in $\mathcal{M}$ and such that every world in W is the denotation of some nominal in S under V. Then $\mathcal{F} \models$ `Axiom`.*

Theorem 5.13 (Completeness and termination) *Let `Axiom` denote a finite set of negative, F-free formulas plus possibly the axiom* (ref)*. The calculus `L` + `Axiom` is terminating and complete with respect to the frames defined by the formulas in `Axiom`.*

Proof. Termination has already been proved (Lemma 5.8). Assume Θ is an open, saturated branch in the calculus `L` + `Axiom`. By Lemma 5.12 it suffices to prove that $\mathcal{M}^\Theta$ is a model satisfying the root formula of Θ and globally satisfying all substitution instances of the formulas in $\texttt{Axiom}$. The fact that $\mathcal{M}^\Theta$ satisfies the root formula follows from the fact that if the root of Θ is $a\phi$ then Θ also contains $u_\Theta(a)\phi$, and thus by Lemma 5.11 we get $\mathcal{M}^\Theta, u_\Theta(a) \models \phi$. That all substitution instances of the formulas in $\texttt{Axiom}$ are globally satisfied in $\mathcal{M}^\Theta$ follows directly from Lemma 5.11. Note that the set of worlds W^Θ of $\mathcal{M}^\Theta$ is the set of urfathers, which is also the image set of the valuation V^Θ of $\mathcal{M}^\Theta$. □

Corollary 5.14 *Let $\texttt{Axiom}$ denote any subset of the following pure formulas:* (ref), (irr), (asym) *and* (intrans). *The calculus $\texttt{L} + \texttt{Axiom}$ is terminating and complete with respect to the frames defined by the formulas in $\texttt{Axiom}$.*

6 Termination with extra side conditions

Let $\texttt{Axiom}$ be a finite set of F-free formulas plus possibly the axiom (ref). We will now prove that by introducing an additional side condition to the (Id) rule, the calculus $\texttt{L} + \texttt{Axiom}$ becomes both terminating and complete with respect to the frame properties defined by $\texttt{Axiom}$. That is, we get a termination and completeness result that in addition to the axioms covered by the result above also covers the non-negative, F-free formulas. Of the axioms in Figure 2 these are (antisym), (uniq) and (tree).

Let (Id^-) denote the following version of the (Id) rule, carrying an extra side condition:

$$\frac{a\phi, ab}{b\phi}\ (Id^-)$$

1. If $a\phi$ is of the form aFc then it is a root subformula.
2. Either ϕ is a nominal or the depth of b is less than or equal to the depth of a.

By $\texttt{L}^-$ we denote the calculus obtained from $\texttt{L}$ by replacing (Id) with (Id^-). Note that Lemmas 5.1, 5.2, 5.3, 5.5 and 5.7 also hold for $\texttt{L}^- + \texttt{Axiom}$ when $\texttt{Axiom}$ is a set of F-free formulas. Furthermore, we have the following result, corresponding to Lemma 5.6. Note, however, that here we need to do induction on the length of Θ; in Lemma 5.6 induction on the depth of nominals sufficed. This makes the proof below slightly more complicated than the corresponding proof of Lemma 5.6. On the other hand, we no longer need an equivalent of Lemma 5.4 for the proof to go through.

Lemma 6.1 (Decreasing length) *Let $\texttt{Axiom}$ be a finite set of F-free formulas, and let Θ be a tableau branch in $\texttt{L}^- + \texttt{Axiom}$. For all nominals a in Θ, $m_\Theta(a) \leq |root_\Theta| - d_\Theta(a)$.*

Proof. The proof is by induction on the length of Θ. The base case where Θ has length 0 is trivial, as Θ then only consists of the root formula. For the induction step assume $\forall a \in Nom_\Theta(m_\Theta(a) \leq |root_\Theta| - d_\Theta(a))$, and let Γ be Θ extended by either one rule application or the introduction of one new axiom. We need to prove that for all a in Nom_Γ:

$$m_\Gamma(a) \leq |root_\Gamma| - d_\Gamma(a). \tag{3}$$

Assume first that Γ differs from Θ by an application of (F). Then Γ is obtained by adding formulas bFc and $c\phi$ where $bF\phi$ belongs to Θ. For all nominals a distinct from c we immediately get (3), using the induction hypothesis. So we only need to prove $m_\Gamma(c) \leq |root_\Gamma| - d_\Gamma(c)$. If $c\phi$ is not a root subformula then $m_\Gamma(c) = -\infty$, and there's nothing to prove. If $c\phi$ is a root subformula then $m_\Gamma(c) = |c\phi|$. Since $c\phi$ was introduced by an application of (F) to $bF\phi$ we get $b \prec_\Gamma c$. This implies $d_\Gamma(c) = d_\Gamma(b) + 1$. We then get, using the induction hypothesis,

$$\begin{aligned} m_\Gamma(c) = |c\phi| = |bF\phi| - 1 \leq m_\Theta(b) - 1 \leq |root_\Theta| - d_\Theta(b) - 1 \\ = |root_\Gamma| - d_\Gamma(b) - 1 = |root_\Gamma| - d_\Gamma(c), \end{aligned} \tag{4}$$

as required. Assume now Γ is *not* obtained from Θ by an application of (F). Then Γ differs from Θ by the addition of a single formula $c\phi$ where c already occurs on Θ. For all a distinct from c we immediately get that (3) holds, using the induction hypothesis. Thus we only need to prove $m_\Gamma(c) \leq |root_\Gamma| - d_\Gamma(c)$. If $m_\Gamma(c) = m_\Theta(c)$ this follows immediately from the induction hypothesis. So assume $m_\Gamma(c) \neq m_\Theta(c)$. Then $c\phi$ is a root subformula and $m_\Gamma(c) = |c\phi|$. Thus $c\phi$ is of maximal length among the root subformulas of the form $c\psi$ on Γ. The formula $c\phi$ can not have been introduced by an application of either $(\vee)$ or $(\wedge)$, since this contradicts maximality. The formula $c\phi$ can not have been introduced by (Nom) either, since in that case $|c\phi| = 2$, and we get $m_\Gamma(c) = m_\Theta(c)$, contradicting the assumption. Thus $c\phi$ must have been introduced by an application of either (@), (G) or (Id^-) to a root subformula. If $c\phi$ is introduced by an application of (@) to a root subformula of the form $bc\phi$ then c is a root nominal, and thus $d_\Gamma(c) = 0$. In this case we trivially get $m_\Gamma(c) \leq |root_\Gamma| - d_\Gamma(c)$. Assume $c\phi$ is introduced by an application of (G) to premises $bG\phi$ and bFc. We can assume that c is a non-root nominal, since otherwise the required conclusion will again follow trivially. Since $c\phi$ is of maximal length we must have $b \neq c$. Lemma 5.5 thus implies $b \prec_\Gamma c$, and therefore $d_\Gamma(c) = d_\Gamma(b) + 1$. We therefore get the same series of inequalities as in (4), except $bF\phi$ is replaced by $bG\phi$. Assume finally that $c\phi$ is produced by applying (Id^-) to premises $b\phi$ and bc. The formula ϕ can not be a nominal, since then $|c\phi| = 2$ and thus $m_\Gamma(c) = m_\Theta(c)$, contradicting the assumption. Since ϕ is not a nominal we must have $d_\Theta(c) \leq d_\Theta(b)$, by side condition 2 of (Id^-). Hence we get, as required, $m_\Gamma(c) = |c\phi| = |b\phi| \leq m_\Theta(b) \leq |root_\Theta| - d_\Theta(b) \leq |root_\Gamma| - d_\Theta(c) = |root_\Gamma| - d_\Gamma(c)$.□

Combining Lemma 6.1 with Lemma 5.7 immediately gives us the following termination result.

Theorem 6.2 (Termination) *Let* `Axiom` *be a finite set of F-free formulas plus possibly the axiom* (ref)*. Any tableau in the calculus* $\mathtt{L}^- + \mathtt{Axiom}$ *is finite.*

6.1 Completeness

To ensure completeness of the calculus $\mathtt{L}^-$ we need to choose the function σ used in the definition of urfathers, Definition 5.9, in a special way. Given a tableau branch Θ and a non-empty set $N \subseteq Nom_\Theta$ we require that $d_\Theta(\sigma_\Theta(N)) = \min\{d_\Theta(a) \mid a \in N\}$. It is obviously possible to choose σ such that this holds. Note that then Lemma 5.10 will still hold with $\mathtt{L}$ replaced by $\mathtt{L}^-$. This implies that Lemma 5.11 will also hold with $\mathtt{L}$ replaced by $\mathtt{L}^-$. Since we have already proven termination, Theorem 6.2, we then get the following results.

Theorem 6.3 (Completeness and termination) *Let* $\mathtt{Axiom}$ *denote a finite set of F-free formulas plus possibly the axiom* (ref)*. The calculus* $\mathtt{L}^- + \mathtt{Axiom}$ *is terminating and complete with respect to the frames defined by the formulas in* $\mathtt{Axiom}$*.*

Corollary 6.4 *Let* $\mathtt{Axiom}$ *denote any subset of the following pure formulas:* (ref), (irr), (asym), (antisym), (intrans), (uniq) *and* (tree)*. The calculus* $\mathtt{L}^- + \mathtt{Axiom}$ *is terminating and complete with respect to the frames defined by the formulas in* $\mathtt{Axiom}$*.*

This result has some interesting applications. For example, it provides a terminating tableau algorithm for simple logics of trees (like those proposed in [5]) for describing the grammatical structure of sentences of ordinary human languages such as English and French. On the other hand, it *doesn't* cover the more complex logics proposed for this purpose in [6], for the simple reason that we still don't know how to handle transitivity. This is the problem to which we now turn.

7 Termination with loop-checks

The axioms of Figure 2 we still haven't been able to deal with are (sym), (trans), (trich), (univ), (serial), (euc) and (dense). The non-termination of the first 4 of these is demonstrated in the Figures 3, 5, 7 and 6. The termination problem demonstrated by these 4 examples is more profound than the one involved in the axioms (antisym), (uniq) and (tree). In the following we will explain why.

Given a tableau branch Θ, define $\sim_\Theta$ to be the reflexive, symmetric and transitive closure of the relation $\{(a,b) \in \mathit{Nom}_\Theta \mid ab \in \Theta\}$. This relation is an equivalence relation by definition. We say that two nominals a and b are *equivalent* on a branch Θ if $a \sim_\Theta b$. The termination problem arising with the axioms (antisym), (uniq) and (tree) is that they allow us to repeatedly introduce new nominals to a branch that are equivalent to already existing nominals on the branch. This can be done *ad infinitum* as shown by the examples in Figures 4 and 8. However, since it is easy to check whether two nominals are equivalent, it is also easy to avoid this kind of non-termination problem. All that is required is to put an extra constraint on the ability to copy formulas between equivalent prefixes. This is the extra constraint we have on the (Id^-) rule compared to the (Id) rule.

In the case of the axioms (sym), (trans), (trich) and (univ) the non-termination problem is less simple. Consider the infinite tableau branches including these axioms in Figures 3, 5, 7 and 6. None of these tableau branches contain pairs of distinct and equivalent nominals, so the source of non-termination can not be equivalence between nominals. Let us define a pair of nominals a, b on a tableau branch Θ to be *twins* if they make the same root subformulas true on Θ, that is, if $\{\phi \mid a\phi \text{ is a root subformula on } \Theta\} = \{\phi \mid b\phi \text{ is a root subformula on } \Theta\}$. The source of non-termination in the case of the axioms (sym), (trans), (trich) and (univ) is that we are allowed to repeatedly introduce new nominals to a branch that are twins of already existing nominals on the branch. This is illustrated by the fact that in all of the tableau branches of Figures 3, 5, 7 and 6, the nominals a_1 and a_2

	1	a_0Fp	
	2	$a_0G(\neg p \vee F(Fp \wedge Gp))$	
	3	$a_0G(q \vee \neg p)$	
	4	a_0Fa_1	(F) on 1
	5	a_1p	(F) on 1
	6	$a_1(\neg p \vee F(Fp \wedge Gp))$	(G) on 2,4
	7	$a_1(q \vee \neg p)$	(G) on 3,4
	8	$a_1F(Fp \wedge Gp)$	$(\vee)$ on 6
	9	a_1q	$(\vee)$ on 7
	10	a_1Fa_2	(F) on 8
	11	$a_2(Fp \wedge Gp)$	(F) on 8
	12	a_2Fp	$(\wedge)$ on 11
	13	a_2Gp	$(\wedge)$ on 11
	14	a_2Fa_3	(F) on 12
	15	a_3p	(F) on 12
	16	$a_0(GG\neg a_2 \vee Fa_2)$	(trans)
	17	a_0Fa_2	$(\vee)$ on 16
	18	$a_2(\neg p \vee F(Fp \wedge Gp))$	(G) on 2,17
	19	$a_2\neg p$	$(\vee)$ on 18
	20	$a_2(q \vee \neg p)$	(G) on 3,17
	21	$a_0(GG\neg a_3 \vee Fa_3)$	(trans)
	22	a_0Fa_3	$(\vee)$ on 21
	23	$a_3(\neg p \vee F(Fp \wedge Gp))$	(G) on 2,22
	24	$a_3F(Fp \wedge Gp)$	$(\vee)$ on 23
	25	$a_3(q \vee \neg p)$	(G) on 3,22
	26	a_3q	$(\vee)$ on 25
	27	$a_1(GG\neg a_3 \vee Fa_3)$	(trans)
	28	a_1Fa_3	$(\vee)$ on 27
	29	$a_i(GG\neg a_j \vee Fa_j)$	(trans) for $j \leq i+1$
	30	$a_iGG\neg a_j$	$(\vee)$ on 29 for $j \leq i+1$
	31	$a_iG\neg a_j$	(G) on 30, $\cdot$ for $j \leq i$
	32	$a_i\neg a_j$	(G) on 31, $\cdot$ for $j < i$
no $(\mathcal{D})$	33	a_3Fa_4	(F) on 24
	34	$a_0(GG\neg a_4 \vee Fa_4)$	(trans)
	35	a_0Fa_4	$(\vee)$ on 34
	36	$a_4(q \vee \neg p)$	(G) on 3,35
	37	$a_4\neg p$	$(\vee)$ on 36
	38	$a_2(GG\neg a_4 \vee Fa_4)$	(trans)
	39	a_2Fa_4	$(\vee)$ on 38
	40	a_4p	(G) on 13,39
		$\times$	

Fig. 9. Incompleteness of $\mathtt{L} + \{(\text{trans})\}$ under the constraint $(\mathcal{D})$.

are twins—and thus all further introduced nominals will be twins of a_1 as well. To ensure termination we can put a constraint on the generation of twin nominals of existing nominals. This is done in [7] by introducing the following general constraint on the construction of tableaus:

$(\mathcal{D})$ The rule (F) is only allowed to be applied to a formula $a\phi$ on a branch Θ if there is no pair of distinct twins $b, c \prec_\Theta^* a$.

This constraint is sufficient to ensure termination, as we will see below. However, now completeness is at stake. In fact, the calculus $\mathtt{L} + \{(\text{trans})\}$ is *not* complete under the constraint $(\mathcal{D})$ as shown by the tableau branch in Figure 9. Lines 1–32 in this figure constitute a saturated tableau branch generated under restriction $(\mathcal{D})$. Under restriction $(\mathcal{D})$ the rule (F) can not be applied to the formula $a_3F(Fp \vee Gp)$ in line 24, since a_1 and a_3 are twins and $a_1 \prec_\Theta^* a_3$. However, if we drop restriction $(\mathcal{D})$, we can continue the branch as in lines 33-40, and the branch closes. This proves that when restriction $(\mathcal{D})$ is applied, the calculus $\mathtt{L} + \{(\text{trans})\}$ is not complete.

The situation we are facing can be summed up as follows: to obtain termination when adding (trans) we need to apply a loop-check condition like $(\mathcal{D})$, but when this is done we loose completeness. Does this mean that we can't deal with transitivity in the present framework? Fortunately it doesn't, but to do it we need to resort to a trick used for standard modal logics, cf. [10]. Instead of using the axiom (trans) we replace the rule (G) by the following rule:

$$\boxed{\dfrac{aG\phi, aFb}{\begin{matrix} b\phi \\ bG\phi \end{matrix}} \; (G_{\text{trans}})}$$

Using this rule instead of (G) and doing loop-check by restriction $(\mathcal{D})$ gives us a terminating calculus with respect to the transitive frames, as we will see below. Since we are anyway forced to do loop-checks we can just as well extend our language to incorporate both the global modality and inverse modalities. Thus, let $\mathtt{L}^+$ be the calculus consisting of the rules $(\neg_1), (\neg_2), (\wedge), (\vee), (F), (G_{\text{trans}}), (@), (Nom), (Id)$ and the following additional rules for the global and inverse modalities:

$$\frac{aE\phi}{b\phi}\ (E)^1 \qquad\qquad \frac{aA\phi}{c\phi}\ (A)^2$$

$$\frac{aP\phi}{\begin{array}{c} bFa \\ b\phi \end{array}}\ (P)^1 \qquad\qquad \frac{aH\phi, bFa}{b\phi}\ (H)$$

[1] The nominal b is new to the tableau.
[2] The nominal c is already on the branch.

We have to take care of the additional rules in our loop-check, so we replace restriction $(\mathcal{D})$ by the following:

$(\mathcal{D}^+)$ The rules (F), (E) and (P) are only allowed to be applied to a formula $a\phi$ on a branch Θ if there is no pair of distinct twins $b, c \prec^*_\Theta a$.

Theorem 7.1 (Completeness and termination) *The calculus $\mathtt{L}^+$ with restriction $(\mathcal{D}^+)$ is terminating and complete with respect to transitive frames.*

Proof. We only give a sketch of the proof. Termination is already proven in [7] with (G) instead of (G_{trans}). Replacing (G) by (G_{trans}) doesn't affect termination, as the termination argument is only based on the fact that the number of root subformulas is finite whenever the number of nominals is finite. To prove completeness we need to construct a model from an open, saturated tableau branch Θ. In [7] it is shown how to construct such a model $\mathcal{M}^\Theta = (\mathcal{W}^\Theta, \mathcal{R}^\Theta, \mathcal{V}^\Theta)$. We can use the same model in the present case, except we replace R^Θ by its transitive closure. In this way transitivity of the model is automatically ensured. The proof that the formulas of Θ hold in the constructed model now follows the same lines as the proof given in [7]. The only non-trivial difference is the case of formulas of the form $G\phi$. This case follows the lines of the proof of completeness for transitive logics in [10]. □

8 Conclusion

In this paper we presented the first systematic results we know of concerning termination of hybrid tableaus for modal and hybrid logics richer than **K**. Much, however, remains to be done—and the work that now faces us seems to vary widely in difficulty. Perhaps the easiest task is to extend the results of the previous section for full nominal tense logic with the universal modality. In this paper we only treated the case of transitivity, but our preliminary investigations suggest that it should be reasonably straightforward to extend the loop-checking approach to cover such conditions as (sym), (trich), (ref) and (trans) in various combinations.

But then we are faced with the task of combining such conditions as (irr), (sym), (asym), (antisym), (intrans), (uniq) and (tree) with (trans), and here matters are likely to be much trickier. Certainly loop-checks will usually be required, but it is unclear to us at present what kinds of general results we can hope for here, or what languages we can prove them for. To give an idea of the difficulties involved, note that even such a simple looking combination as (trans)+(irr) does not have the finite frame property (consider the formula $F\top \wedge GF\top$, for example). There are some natural ways around such difficulties. For example, as is noted in [2], hybrid logics lacking the finite *frame* property often have the finite *model* property with respect to non-standard but relatively simple classes of models. But speculation before detailed investigations have been carried out is probably fruitless here: in termination proofs, the devil is in the details.

References

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

[2] Patrick Blackburn. Nominal tense logic. *Notre Dame Journal of Formal Logic*, 34(1):56–83, 1993.

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

[4] Patrick Blackburn, Marrten de Rijke, and Yde Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 2001.

[5] Patrick Blackburn, Claire Gardent, and Wilfried Meyer-Viol. Talking about trees. In *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–29, 1993.

[6] Patrick Blackburn and Wilfried Meyer-Viol. Linguistics, logic, and finite trees. *Logic Journal of the IGPL*, 2:3–29, 1994.

[7] Thomas Bolander and Patrick Blackburn. Termination for hybrid tableaus. *Journal of Logic and Computation*, 17(3):517–554, 2007.

[8] Thomas Bolander and Torben Braüner. Tableau-based decision procedures for hybrid logic. *Journal of Logic and Computation*, 16:737–763, 2006.

[9] George Gargov and Valentin Goranko. Modal logic with names. *Journal of Philosophical Logic*, 22(6):607–636, 1993.

[10] Rajeev Gore. Tableau methods for modal and temporal logics. Technical Report TR-ARP-15-95, Australian National University, November 1995.

[11] Balder ten Cate and Massimo Franceschet. On the complexity of hybrid logics with binders. In *Proceedings of Computer Science Logic 2005*, volume 3634 of *Lecture Notes in Computer Science*, pages 339–354. Springer Verlag, 2005.

[12] Miroslava Tzakova. Tableau calculi for hybrid logics. *Lecture Notes in Computer Science*, 1617:278–292, 1999.