UNIVERSITÀ DEGLI STUDI DI ROMA TRE
Dipartimento di Informatica e Automazione
Via della Vasca Navale, 79 – 00146 Roma, Italy.

Tableaux with Substitution for Hybrid Logic with the Global and Converse Modalities

SERENELLA CERRITO[1] AND MARTA CIALDEA MAYER[2]

RT-DIA-155-2009 **November 2009**

[1] Ibisc, FRE CNRS 3190, Université d'Evry Val d'Essonne
[2] Università di Roma Tre, Italy

ABSTRACT

This work provides the full proofs of the properties of the tableaux calculus for hybrid logic with the global and converse modalities presented in [3], which focuses on the $HL(@)$ fragment of the calculus. While such a fragment terminates without loop checks, when the converse and global modalities are added to the language, and the corresponding rules to the system, termination is achieved by means of a loop checking mechanism. The peculiarity of the system is the treatment of nominal equalities by means of a substitution rule. The main advantage of such a rule, compared with other approaches, is its efficiency, that has been experimentally verified for the $HL(@)$ fragment. Such an advantage should persist in the extended calculus.

In this work we give the detailed termination and completeness proofs for the entire calculus. Although the main guidelines are the same as the corresponding proofs for $HL(@)$, the proofs for the extended calculus conceal many subtleties that have to be handled with care.

1 Introduction

Hybrid logic extends modal logic with the possibility of naming worlds by means of *nominals*, i.e. atomic formulae which hold in exactly one world. *Satisfaction assertions* are formulae of the form $@_a F$ (where @ is the satisfaction operator, a is a nominal and F a formula), stating that F holds at the world named a. The satisfaction operator is the only additional operator, with respect to standard modal logic K, of the so called *basic* hybrid logic, $HL(@)$.

A previous work, [4], further refined and extended in [3], presented a tableaux calculus for $HL(@)$, called H, whose characterizing feature is the treatment of *nominal equalities*, i.e. formulae of the form $@_a b$, stating that a and b actually name the same world, by means of a substitution rule. Such a rule minimizes possible redundancies deriving from the presence of nominal equalities. Like other calculi for basic hybrid logic, H enjoys strong termination (every tableau in H is finite, independently of the rule application order), and termination does not need loop checks. Moreover, the system does not use any extra-logical notation, like prefixes, i.e. it is an *internalized* calculus. The computational advantages of substitution, with respect to other approaches (mainly [2]), has been experimentally verified [5].

In [3], the extension of H handling the global (A and E) and converse ($\Box^-$ and $\Diamond^-$) modalities is presented. The expansion rules added to deal with the new operators are the same as those proposed in [1] and, like in [2, 1], termination is achieved by means of a loop checking mechanism. The extended calculus, which will be called H^+, should share with H its computational advantages, since it preserves the substitution rule.

In [3], the proofs of the fundamental properties of H^+, namely soundness, completeness and termination, are given in detail only for the H subsystem. This work gives the full proofs for H^+.

2 Preliminaries

This section contains the basic definitions that will be used in the sequel, concerning the syntax and semantics of the "uni-modal" version of $HL(@)$, its extension to the multimodal case being straightforward.

Let NOM and PROP be disjoint sets of propositional letters. The elements of NOM are called *nominals* and the elements of $\mathsf{NOM} \cup \mathsf{PROP}$ *atoms*. We shall use lowercase letters from the beginning of the alphabet, possibly with indexes, as metavariables for nominals, and p, q, r, possibly with indexes, for elements of PROP. The set of formulae in $HL(@)$ is defined by the following grammar:

$$F := \bot \mid p \mid a \mid \neg F \mid F \wedge F \mid F \vee F \mid @_a F \mid \Box F \mid \Diamond F \mid \Box^- F \mid \Diamond^- F \mid AF \mid EF$$

where $p \in \mathsf{PROP}$ and $a \in \mathsf{NOM}$.

An *interpretation* $\mathcal{M}$ is a quadruple $\langle W, R, N, I\rangle$ where W is a non-empty set (whose elements are the *states* of the interpretation), $R \subseteq W \times W$ (the *accessibility relation*), N is a function $\mathsf{NOM} \rightarrow \mathsf{W}$ and I a function $W \rightarrow 2^{\mathsf{PROP}}$. We shall write wRw' as a shorthand for $\langle w, w'\rangle \in R$.

If $\mathcal{M} = \langle W, R, N, I\rangle$ is an interpretation, $w \in W$ and F a formula, the relation $\mathcal{M}_w \models F$ ($\mathcal{M}$ satisfies F at w) is inductively defined as follows:

1. $\mathcal{M}_w \not\models \bot$.
2. $\mathcal{M}_w \models p$ if $p \in I(w)$, for $p \in \mathsf{PROP}$.
3. $\mathcal{M}_w \models a$ if $N(a) = w$, for $a \in \mathsf{NOM}$.
4. $\mathcal{M}_w \models \neg F$ if $\mathcal{M}_w \not\models F$.
5. $\mathcal{M}_w \models F \wedge G$ if $\mathcal{M}_w \models F$ and $\mathcal{M}_w \models G$.
6. $\mathcal{M}_w \models F \vee G$ if either $\mathcal{M}_w \models F$ or $\mathcal{M}_w \models G$.
7. $\mathcal{M}_w \models @_a F$ if $\mathcal{M}, N(a) \models F$.
8. $\mathcal{M}_w \models \Box F$ if for each w' such that wRw', $\mathcal{M}_{w'} \models F$.
9. $\mathcal{M}_w \models \Diamond F$ if there exists w' such that wRw' and $\mathcal{M}_{w'} \models F$.
10. $\mathcal{M}_w \models \Box^- F$ if for each w' such that $w'Rw$, $\mathcal{M}_{w'} \models F$.
11. $\mathcal{M}_w \models \Diamond^- F$ if there exists w' such that $w'Rw$ and $\mathcal{M}_{w'} \models F$.
12. $\mathcal{M}_w \models AF$ if for each $w' \in W$, $\mathcal{M}_{w'} \models F$.
13. $\mathcal{M}_w \models EF$ if there exists $w' \in W$ such that $\mathcal{M}_{w'} \models F$.

A formula F is *satisfiable* if there exist an interpretation $\mathcal{M}$ and a state w of $\mathcal{M}$, such that $\mathcal{M}_w \models F$. Two formulae F and G are logically equivalent ($F \equiv G$) iff for every interpretation $\mathcal{M}$ and state w of $\mathcal{M}$, $\mathcal{M}_w \models F$ if and only if $\mathcal{M}_w \models G$.

It is worth pointing out that, for any nominal a and formula F:

$$\neg @_a F \equiv @_a \neg F \qquad \neg \Diamond F \equiv \Box \neg F \qquad \neg \Box F \equiv \Diamond \neg F$$
$$\neg \Diamond^- F \equiv \Box^- \neg F \qquad \neg \Box F^- \equiv \Diamond^- \neg F \qquad \neg AF \equiv E \neg F \qquad \neg EF \equiv A \neg F$$

This allows one to restrict attention to formulae in negation normal form (where negation dominates only atoms), without loss of generality.

3 The tableau calculus $\mathbf{H}^+$

Tableau nodes are labelled by sets of *satisfaction statements*, i.e. assertion of the form $@_a F$. A formula of the form $@_a F$ will be called *labelled by a*. If $@_a F \in S$, where S is a tableau node, we say that F is true at a in S. A formula of the form $@_a \Diamond b$, where b is a nominal, is a *relational formula.*

In the sequel, sets of formulae will be written as comma separated sequences of formulae. For the sake of simplicity, we assume that formulae are in negation normal form (nnf).

The initial tableau for a set S of formulae is a node labelled by $S_a = \{@_a F \mid F \in S\}$, where a is a new nominal. S_a is called the *root set.* Nominals occurring in S_a are called *root nominals*, and, if T is a tableau rooted at S_a, then the set of its root nominals is denoted by C_T:

$$C_T = \{b \mid b \text{ is a nominal occurring in } S_a\}$$

Boolean Rules
$$\frac{@_a(F \wedge G), S}{@_a F, @_a G, @_a(F \wedge G), S}\ (\wedge)$$ $$\frac{@_a(F \vee G), S}{@_a F, @_a(F \vee G), S \qquad @_a G, @_a(F \vee G), S}\ (\vee)$$

Label Rule
$$\frac{@_a @_b F, S}{@_b F, @_a @_b F, S}\ (@)$$

Modal rules	
$$\frac{@_a \Box F, @_a \Diamond b, S}{@_b F, @_a \Box F, @_a \Diamond b, S}\ (\Box)$$	$$\frac{@_a \Diamond F, S}{@_a \Diamond b, @_b F, @_a \Diamond F, S}\ (\Diamond)$$ where b is a new nominal (not applicable if F is a nominal)
Closure rules	
$$\frac{@_a p, @_a \neg p, S}{\bot}\ (\bot_1)$$	$$\frac{@_a \neg a, S}{\bot}\ (\bot_2)$$

Table 1: Logical and closure rules for $HL(@)$.

Table 1 contains the *logical* rules, i.e. all rules but substitution, of the system H for basic hybrid logic, $HL(@)$, already presented in [4, 3]. The additional rules of H^+ for the global and converse modalities, added in [3], are given in Table 2.

The $\Diamond$, $\Diamond^-$ and E rules are called *nominal generating rules*.

It is worth pointing out that, contrarily to the $\Diamond$-rule, there is no restriction on the applicability of the $\Diamond^-$-rule; in fact, it is necessary to expand formulae of the form $@_a \Diamond^- c$ where $c \in Nom$, in order to obtain possible premises for the $\Box$ and $\Box^-$-rules, of the form $@_c \Diamond a$.

A tableau node S is *closed* if it contains $\bot$ (see the Closure Rules of Table 1). A tableau branch is open if all its nodes are open (otherwise it is closed). A tableau is closed if all its branches are closed, otherwise it is open.

Definition 1 *A formula occurring in a tableau node is an* accessibility formula *if it is a relational formula introduced by application of the $\Diamond$ or $\Diamond^-$-rule.*

A formula occurring in a tableau T is called native *(in T) iff it is in the language of the root set, i.e. it does not contain any non-root nominal.*

Note that accessibility formulae are not native.

Converse rules	
$\dfrac{@_a\Box^- F, @_b\Diamond a, S}{@_b F, @_a\Box^- F, @_b\Diamond a, S}\ (\Box^-)$	$\dfrac{@_a\Diamond^- F, S}{@_b\Diamond a, @_b F, @_a\Diamond^- F, S}\ (\Diamond^-)$ where b is a new nominal
Global Rules	
$\dfrac{@_a AF, S}{@_c F, @_a AF, S}\ (A)$ where c occurs in the premise	$\dfrac{@_a EF, S}{@_b F, @_a EF, S}\ (E)$ where b is a new nominal

Table 2: Rules for the converse and global modalities

In order to define the last rule of the system, the substitution rule, the definition of father and children of a nominal, given in [4, 3], has to be extended.

Definition 2 *Let Θ be a tableau branch. If one of the nominal-generating rules ($\Diamond$, $\Diamond^-$ or E) has been applied in Θ to a formula $@_a F$ generating a new nominal b, then $a \prec_\Theta b$ (and we say that b is a child of a, and a is the father of b).*

The relation $\prec_\Theta^+$ is the transitive closure of $\prec_\Theta$ and $\prec_\Theta^*$ the reflexive and transitive closure of $\prec_\Theta$. If $a \prec_\Theta^+ b$ we say that b is a descendant of a and a an ancestor of b in the branch Θ.

The substitution rule, which is applicable only if $a \neq b$, is formulated as follows:

$$\frac{@_a b, S}{S^\#[a \mapsto b]}\ (Sub)$$

where $S^\#[a \mapsto b]$ is obtained from S by:

1. deleting every formula containing a descendant of a;

2. replacing every occurrence of a with b.

When the substitution rule is applied, a is said to be *replaced in the branch* and the descendants of a are called *deleted in the branch.*

Like in [4, 3], trivial cases of non-termination are ruled out by the following restriction: a formula is never added to a node where it already occurs and the nominal-generating rules are never applied twice to the same premise on the same branch.

Termination relies on loop-checking, that, like in [1], exploits the notion of *twin nominals*:

Definition 3 *Let T be a tableau, Θ a branch of T and S a node of Θ.*

Then:

- *If a is a nominal occurring in S then*

$$Forms_S(a) = \{F \mid a : F \textit{ occurs in } S \textit{ and } F \textit{ is native in } T\}$$

i.e. $Forms_S(a)$ contains all the native formulae labelled by a in S.

- *Two nominals a and b are said to be twins in S if $Forms_S(a) = Forms_S(b)$.*

- *If a is a nominal occurring in S, a is an* urfather *in S if there is no pair of distinct twins $b, c \in S$ such that $b, c \prec^*_\Theta a$.*[1]

In other terms, a and b are twins in a tableau node S if they label exactly the same set of native formulae, and a nominal a is an urfather if neither a is a twin of one of his ancestors, nor it is a descendant of two distinct twin nominals. Note that any root nominal is necessarily an urfather, since it has no ancestors at all.

Termination is ensured by the following restriction:

R *The rules $\Diamond, \Diamond^-$ and E are only allowed to be applied to a formula $@_aF$ of a node S on a branch Θ if a is an urfather in S.*

Nominals which are not urfathers in S are said to be blocked in S. A tableau branch is said to be *complete* if no rule can be applied to expand it further (possibly because of restriction **R**).

It is worth pointing out that, so far, the only significant difference with respect to the internalized calculus proposed in [1] is the substitution rule. In fact all the other rules and the blocking mechanism are essentially the same. Obviously, since the substitution rule is destructive and tableaux are to be formulated in the "nodes as sets" style, the notions of twin nominals and urfathers are relative to a single tableau node and not to tableau branches.

4 Properties of $\mathbf{H}^+$

The soundness argument for H^+ runs exactly as for H (see [4, 3]), modulo a previous trivial argument establishing that also the new rules are locally sound. Here, we give the details of the termination and completeness proofs.

4.1 Termination of $\mathbf{H}^+$

In order to show that the extension of H to the global and converse modalities terminates with loop-checks, one can use an argument similar to the one given in [2] for the prefixed calculus. However, the substitution rule has to be handled with care. In fact, when Sub is either applied to $@_ab$, or it substitutes an ancestor of a, the nominal a disappears, so that some arguments are needed in order to show that this does not result in the possibility of expanding a nominal that would otherwise be blocked by a.

Some of the intermediate results proved for H generalize to H^+. In some cases, however, such results are weaker, sometimes they must be strengthened and sometimes the proofs are somewhat subtler. In order to properly state them, we need the following definition:

[1] The expression "urfather" is borrowed from [2].

Definition 4 *Let T be a tableau rooted at S_0, Θ a branch of T, and S a node of Θ. Then:*

$$S^* = \{F \mid F = G[b_1 \mapsto c_1, ..., b_n \mapsto c_n] \text{ for some subformula } G \text{ of some } A \in S, \text{ and } c_1, ..., c_n \in C_T\}$$

where $b_1, \cdots, b_n$ are nominals occurring in G. In particular, the set S_0^* contains every formula that can be obtained from a subformula of some formula in the initial set, by replacing nominals with other nominals still occurring in the initial set.

Note that S_0^* is necessarily finite.

In order to ensure termination with loop checks, the *quasi-subformula* property of H has to be strengthened.

Lemma 1 (Quasi-subformula property) *If T is a tableau rooted at S_0, and $@_a F$ is a formula occurring in some node of T, then either $@_a F$ is a relational formula or $F \in S_0^*$.*

Moreover, any tableau branch Θ contains a finite number of native formulae labelled by the same nominal. In particular, for every nominal a, the set of native formulae labelled by a in Θ is a subset of

$$S_0^* \cup \{\Diamond b \mid b \in C_T\}$$

Proof. For the first assertion, the inductive proofs given in [4, 3] can easily be extended to handle the new rules.

In order to prove the second assertion, we must prove that, for any fixed nominal a, the number of relational formulae of the form $@_a \Diamond b$ where b is a root nominal is finite. But this is straightforward, since there is only a finite number of root nominals.

The following properties are direct consequences of Lemma 1: if T is a tableau rooted at S_0, then:

1. If $@_a b$ occurs in a node of T, then $b \in C_T$. Therefore, in the applications of the substitution rule, nominals are always replaced by root nominals.
2. A nominal $b \notin C_T$ may occur in tableau nodes only in relational formulae of the form $@_a \Diamond b$ or $@_b \Diamond a$, or as the label of a formula $@_b F$, where $F \in S_0^*$ contains only root nominals.

Lemma 2 *If c and d are nominals, $c, d \notin C_T$, and $@_c \Diamond d$ occurs in some node of a tableau branch Θ, then either $c \prec_\Theta d$ or $d \prec_\Theta c$, i.e. $@_c \Diamond d$ is an accessibility formula.*

Proof. If $c, d \notin C_T$, then $@_c \Diamond d$ cannot be obtained by application of the substitution rule from an accessibility formula, since only root nominals can replace other nominals.

The above result is weaker than the corresponding one holding for H. In fact, if $@_c \Diamond d$ is an accessibility formula, it can be introduced either by the $\Diamond$ or the $\Diamond^-$ rule, so that it can be either $c \prec_\Theta d$ or $d \prec_\Theta c$. Moreover, if c is a root nominal, $@_c \Diamond d$ can be obtained by substitution from an accessibility formula $@_a \Diamond d$, and, if $d \prec_\Theta a$, d is not deleted when replacing a with c.

From the above results it follows that if $@_a F$ occurs in a tableau, $a \notin C_T$ and F contains any non-root nominal, then it is an accessibility formula.

Moreover:

Lemma 3 *If Θ is a tableau branch and a any nominal occurring in Θ, then $\{b \mid a \prec_\Theta b\}$ is finite.*

Moreover, in any tableau branch the number of accessibility formulae labelled by the same nominal a is finite.

Proof. The first assertion is obvious, since the number of expandable formulae labelled by the same nominal a is finite and the nominal-generating rules are never applied twice to the same formula in a branch.

For the second assertion, let a be any fixed nominal occurring in a branch Θ and $b_0, b_1, ...$ all the nominals such that $@_a\Diamond b_i$ is an accessibility formula occurring in Θ. Each of them can be introduced either expanding a formula of the form $@_a\Diamond F$ (where F is not a nominal) or a formula of the form $@_{b_i}\Diamond^- F$. In other terms, for all b_i, either $a \prec_\Theta b_i$ or $b_i \prec_\Theta a$.

By Lemma 1, there is only a finite number of expandable formulae of the form $@_a\Diamond F$, hence the number of new nominals b_i such that $a \prec_\Theta b_i$ is finite.

The expansion of a formula of the form $@_{b_i}\Diamond^- F$ generates a as a new nominal, therefore there can be at most one accessibility formula of the form $@_a\Diamond b_i$ with $b_i \prec a$.

From Lemmas 1, 2 and 3, it follows that:

Corollary 1 *For every tableau branch Θ and nominal a, if $a \notin C_T$, then the set*

$$\{F \mid @_aF \textit{ occurs in } \Theta\}$$

is finite.

If $a \in C_T$ the set

$$\{F \mid @_aF \textit{ occurs in } \Theta \textit{ and } F \textit{ does not have the form } \Diamond b \textit{ for some } b \notin C_T\}$$

is finite.

Note that, differently from the corresponding result holding for H, the above corollary distinguishes between root and non-root nominals. In fact, if it were possible that an infinite number of nominals occurred in a branch, a root nominal might label an infinite number of relational formulae.

Lemma 4 *Let Θ be a tableau branch. If Θ is infinite then there is an infinite chain of nominals*

$$b_1 \prec_\Theta b_2 \prec_\Theta b_3 ... \, .$$

Proof. The presence of the substitution rule makes the argument a little more complicated than the corresponding one given in [2].

First of all we prove that if Θ is infinite, then there is an infinite number of nominals occurring in Θ. If there were only a finite number of nominals, then any nominal, including root nominals, would label a finite number of formulae (Corollary 1). Now, since formulae are never added to nodes where they already occur, there should be at least a formula F occurring in a node S_i of Θ which disappears from the branch and then reappears in a node S_j below S_i. F can disappear only because some nominal occurring in it is either replaced or deleted. But when a nominal is replaced or deleted, it can never occur again in the branch below the application of Sub which replaces/deletes it.

Now, like in [2], we can prove that the infinite number of nominals occurring in Θ can be arranged by $\prec_\Theta$ in a forest of finitely branching trees (because any nominal can generate only a finite number of new ones, by Lemma 3), each of them rooted at a root-nominal. By König's Lemma, if one of such trees is infinite, it has an infinite branch, i.e. there is an infinite chain of nominals $b_1 \prec_\Theta b_2 \prec_\Theta b_3 \ldots$.

Theorem 1 (Termination) *Every tableau is finite.*

Proof. By Lemma 4, if an infinite branch Θ exists, then there is an infinite chain of nominals

$$b_1 \prec_\Theta b_2 \prec_\Theta b_3 \ldots .$$

By Lemma 1, if $@_a F$ occurs in Θ and F is native, then F is an element of the finite set $S_0^* \cup \{\Diamond b \mid b \in C_T\}$, where S_0 is the root set.

Let n be the cardinality of $S_0^* \cup \{\Diamond b \mid b \in C_T\}$ and consider the initial sub-chain:

$$b_1 \prec_\Theta b_2 \prec_\Theta b_3 \ldots b_{2^n+1} \prec_\Theta b_{2^n+2}$$

Let Θ' be the initial segment of Θ up to, but not including, the nominal-generating inference ($\Diamond$, $\Diamond^-$ or E) producing b_{2^n+2}. Let S_k be the last node of Θ'.

Since b_{2^n+1} occurs in S_k, all its ancestors occur in S_k, too, because if some of them had been either replaced or deleted above S_k, b_{2^n+1} would have been deleted by the same application of the substitution rule. Since b_{2^n+1} is the father of b_{2^n+2} in Θ, and it generates b_{2^n+2} by expanding S_k, then b_{2^n+1} is an urfather in S_k, i.e. S_k does not contain two distinct twins $b_i, b_j \prec^* b_{2^n+1}$, otherwise b_{2^n+1} would be blocked in S_k.

Because of the choice of n, however, at least two nominals b_i and b_j among $b_1, \ldots, b_{2^n+1}$ must be twins in S_k (i.e. they must label the same set of native formulae).

4.2 Completeness of $\mathbf{H}^+$

In this section we prove that if Θ is a complete and open branch of a H^+ tableau rooted at S_0, then S_0 is satisfiable.

The proof is similar to the completeness proof of H given in [4, 3] and exploits the termination of H^+. However, like in the case of termination, the presence of the new rules requires attention.

We first consider the set labelling the last node of Θ, that is *downward saturated* (in some sense), and we show that any such set has a model. The model construction in this base step is different from [4, 3], since saturation only affects urfathers. The lifting result, showing that satisfiability propagates upward to the root node, is proved like in [4, 3].

In what follows, we say that a formula F occurs in Θ (and Θ contains F) to mean that F occurs in some node of Θ.

Here follows the notion of saturation, that is relative to a tableau node, since clauses 8–10 refer to urfathers in order to take into account the blocking mechanism. The notion of urfather, however, depends on the branch, since the relation $\prec_\Theta^*$ is branch-dependant.

Definition 5 *A node S of a tableau branch Θ is* downward saturated *iff the following conditions hold:*

1. *S does not contain any formula of the form $@_a\neg a$, and it does not contain two formulae of the form $@_a p$ and $@_a\neg p$ for some atom p.*
2. *If $@_a(F \wedge G) \in S$, then $@_a F, @_a G \in S$.*
3. *If $@_a(F \vee G) \in S$, then either $@_a F \in S$ or $@_a G \in S$.*
4. *If $@_a @_b F \in S$, then $@_b F \in S$.*
5. *If $@_a\Diamond b, @_a\Box F \in S$, then $@_b F \in S$.*
6. *If $@_b\Diamond a, @_a\Box^- F \in S$, then $@_b F \in S$.*
7. *If $@_a AF \in S$, then for all nominals b occurring in S, $@_b F \in S$.*
8. *If $@_a\Diamond F \in S$, F is not a nominal, and a is an urfather in S, then there is a nominal b such that $@_a\Diamond b, @_b F \in S$.*
9. *If $@_a\Diamond^- F \in S$ and a is an urfather in S, then there is a nominal b such that $@_b\Diamond a, @_b F \in S$.*
10. *If $@_a EF \in S$ and a is an urfather in S, then there is a nominal b such that $@_b F \in S$.*
11. *If $@_a b \in S$ then $a = b$.*

Since any branch Θ is finite by Theorem 1, $\Theta = S_0, S_1, ..., S_k$ for some k. If Θ is open and complete, then S_k is downward saturated.

Definition 6 *Let Θ be a tableau branch, S a node of Θ and b a nominal occurring in S. The urfather of b in S, written $u_S(b)$, is the nominal $a \prec^*_\Theta b$ such that a is a twin of b and a is an urfather, if it exists (undefined otherwise).*

Note that $u_S(b)$ may be undefined. Consider in fact a situation where $a_1 \prec^+_\Theta a_2 \prec^+_\Theta b$ and a_1 and a_2 become twins after the generation of b (by effect of the converse rules). It may happen that, in the chain leading to b, there is no ancestor of b that is a twin of b (because of different choices in the expansion of disjunctive formulae). In such cases, b is not an urfather and $u_S(b)$ does not exist either. It is worth pointing out also that there is at most one urfather $a \prec^*_\Theta b$ that is a twin of b. In fact, if a_1 and a_2 are distinct nominals such that $a_1 \prec^*_\Theta b$, $a_2 \prec^*_\Theta b$ and both a_1, a_2 are twins of b, then a_1 is also a twin of a_2, hence at least one among a_1, a_2 has a twin ancestor and is not an urfather.

The following results establish useful properties of urfathers.

Lemma 5 *For any nominal a occurring in a tableau node S, a is an urfather in S if and only if $u_S(a) = a$.*

Proof. If a is an urfather, then a itself is a nominal meeting all the requirements to be $u_s(a)$ (and such a nominal is unique, as already observed). The converse implication is trivial: for any nominal b, the nominal $u_s(b)$ is an urfather by definition, hence, when $u_S(b) = b$, b is necessarily an urfather.

Lemma 6 *Let Θ be a branch of a tableau, S a node of Θ, a an urfather on S and b a nominal occurring in S. If either $a \prec_\Theta b$ or $b \prec_\Theta a$, then $u_S(b)$ is defined.*

Proof. If $b \prec_\Theta a$ and a is an urfather in S, then necessarily also b is an urfather in S: if there were two twins $c_1, c_2 \prec^*_\Theta b$ then c_1 and c_2 would be twin ancestors of a, so a could not be an urfather. Hence $u_S(b)$ is defined by Lemma 5.

If $a \prec_\Theta b$ and $u_S(b)$ is undefined, then b is not an urfather, otherwise, by the previous lemma, $u_S(b) = b$. Therefore b has at least two twin ancestors. Let c_1 and c_2 be the *first* two nominals, in the generation chain leading to b, such that c_1 and c_2 are twins and $c_1, c_2 \prec^*_\Theta b$; say that $c_1 \prec^+_\Theta c_2 \prec^*_\Theta b$. (note that c_1, c_2, a and b are necessarily on the same chain). Obviously c_1 is an urfather. Moreover, $c_1, c_2 \in S$, because, if any of them had been either replaced or deleted by the *Sub* rule, then also b would be so. If $c_2 \prec^*_\Theta a$, then a would not be an urfather, thus this case is ruled out. If $c_1 \prec^*_\Theta a \prec^*_\Theta c_2$, then necessarily $c_2 = b$, hence $c_1 = u_S(b)$ (because c_1 would be a twin of b and an urfather), contradicting the hypothesis that $u_S(b)$ is undefined. Finally, the case where $a \prec^+_\Theta c_1 \prec^+_\Theta c_2$ is impossible because c_1 and c_2 are distinct, $c_1, c_2 \prec^*_\Theta b$, and $a \prec_\Theta b$. Thus, $u_S(b)$ is necessarily defined.

Lemma 7 *Let S be a node in a tableau rooted at S_0. If $u_S(a) = b$ and F is native, then $@_aF \in S$ if and only if $@_bF \in S$.*

Proof. If $@_aF \in S$ and F is native, then $F \in Forms_S(a)$. The thesis then follows because $u_s(a)$ is a twin of a in S. The reasoning is the same for the other direction of the implication.

The following lemma defines a model for the leaf S of any complete and open branch. Note that S necessarily contains at least one root nominal a_0, which is an urfather in S, and, by Lemma 5, $u_S(a_0) = a_0$.

Lemma 8 *Let S be a saturated and open tableau leaf, and let a_0 be any root nominal occurring in S. Let $\mathcal{M}^*$ be the interpretation defined as follows:*

$W = \{a \mid a \textit{ is an urfather in } S\}$;
$R = \{(u_S(a), u_S(b)) \mid @_a\Diamond b \in S \textit{ and both } u_S(a), u_S(b) \textit{ are defined}\}$;
For every nominal a occurring in S: $N^*(a) = \begin{cases} u_S(a) & \textit{if } u_S(a) \textit{ is defined} \\ a_0 & \textit{otherwise} \end{cases}$
$I(a) = \{p \mid @_ap \in S\}$ *for all* $a \in W$

If $a \in W$, $@_aF \in S$ and F has not the form $\Diamond b$ for some b such that $u_S(b)$ is undefined, then $\mathcal{M}^*_a \models F$.

Proof. We remark beforehand that:

A. If $a \in W$, then $u_S(a)$ is defined and $u_S(a) = a$ (by Lemma 5). Therefore $N^*(a) = a$.

B. If $a \in W$ and $@_aF \in S$, with F having one of the following forms:

   (i) $\Diamond G$ where G is not a nominal,

(ii) $\Diamond^- G$

(iii) EG,

then there is a formula $@_b G \in S$ such that $u_S(b)$ is defined and:

- if $F = \Diamond G$, then also $@_a \Diamond b \in S$;
- if $F = \Diamond^- G$, then also $@_b \Diamond a \in S$.

In fact, since S is saturated and a is an urfather, S contains also some $@_b G$ (and either $@_a \Diamond b$ or $@_b \Diamond a$ in cases (i) and (ii), respectively). Now, such formula (or formulae) cannot have been obtained by replacing a for a nominal c, because in that case the descendants of c have been deleted, so that $@_a F$ has to be expanded again. This expansion of $@_a F$ has generated some $@_d G$ with $a \prec_\Theta d$ (and either $@_a \Diamond d$ or $@_d \Diamond a$ in cases (i) and (ii), respectively). If afterwords b has replaced d, then b is a root nominal. Any other substitution affects both G and F. Therefore, either $a \prec_\Theta b$ or b is a root nominal. In the first case, $u_S(b)$ is defined by Lemma 6, in the second by Lemma 5.

Let us assume that $a \in W$ and $@_a F \in S$, for $F \neq \Diamond b$ with $u_S(b)$ undefined. The proof that $\mathcal{M}^*{}_a \models F$ is by induction on F.

Base We distinguish three cases.

1. F is a literal. If F is a propositional letter or its negation, then the result is true by construction of $\mathcal{M}^*$. In fact, if $@_a p \in S$, for $p \in \mathsf{PROP}$, since $N^*(a) = a$ because $a \in W$ (by remark A), by definition $\mathcal{M}^*_a \models p$. If $@_a \neg p \in S$, since S is open, $@_a p \notin S$ and again $\mathcal{M}^*_a \models \neg p$ by construction.
2. F is a nominal b. Then necessarily $b = a$, since S is saturated, and the result is trivial, since $N^*(a) = a$ (by remark A).
3. F is $\neg b$, for some nominal b. Since S is open, $b \neq a$. By Lemma 1, F is native. Therefore b is a root nominal, hence an urfather, and an element of W, so that $N^*(b) = b \neq a$. Therefore $\mathcal{M}^*_a \models \neg b$.

Induction Step We distinguish several cases according to the form of F.

1. $F = G \wedge H$. If $@_a G \wedge H \in S$, since S is saturated, both $@_a G$ and $@_a F$ are in S. By the inductive hypothesis, $\mathcal{M}^*, a \models G$ and $\mathcal{M}^*, a \models H$, hence $\mathcal{M}^*, a \models G \wedge H$. If F is a disjunction the reasoning is similar.
2. $F = \Box G$. Let b be any element of W such that aRb. By definition, there are two nominals c and d such that $a = u_S(c)$, $b = u_S(d)$ and $@_c \Diamond d \in S$. Since a and c are twins and $\Box G$ is native, $@_c \Box G \in S$ by Lemma 7. And since S is saturated, $@_d G \in S$, so that also $@_b G \in S$ because b and d are twins and G is native (Lemma 7 again). By the inductive hypothesis, then $\mathcal{M}^*_b \models G$. Therefore $\mathcal{M}^*_a \models \Box G$.
3. $F = \Box^- G$. This case is quite similar to the previous one. Let b be any element of W such that bRa. By definition, there are two nominals c and d such that $a = u_S(c)$, $b = u_S(d)$ and $@_d \Diamond c \in S$. Since a and c are twins and $\Box^- G$ is

native, $@_c\Box^- G \in S$ by Lemma 7. And since S is saturated, $@_d G \in S$, so that also $@_b G \in S$ because b and d are twins and G is native (Lemma 7 again). By the induction hypothesis, then $\mathcal{M}_b^* \models G$. Therefore $\mathcal{M}_a^* \models \Box^- G$.

4. $F = AG$. If $@_a AG \in S$ and S is saturated, for any nominal b occurring in S, $@_b G \in S$, hence in particular this holds for urfathers. By the induction hypothesis, for all $b \in W$, $\mathcal{M}_b^* \models G$. Therefore $\mathcal{M}_a^* \models AG$.
5. $F = EG$. If $@_a EG \in S$, since $a \in W$, then there is a formula $@_b G \in S$ such that $u_S(b)$ is defined (by Remark B). Let then $c = u_S(b)$ (thus $c \in W$). Since G is native, $@_c G \in S$ by Lemma 7, and by the inductive hypothesis $M_c \models G$. Therefore $\mathcal{M}_a^* \models EG$.
6. F is $\Diamond G$. We distinguish two cases.
   (a) G is a nominal b. By hypothesis, $u_S(b)$ is defined and belongs to W. So, let $u_S(b) = c$. By construction of $\mathcal{M}^*$, we have: $N^*(a) = a$, $N^*(b) = u_S(b) = c$ and aRc. Hence $\mathcal{M}_a^* \models \Diamond b$.
   (b) G is not a nominal, and therefore it is native. By Remark B, since $a \in W$, there is a nominal b such that $@_a\Diamond b, @_b G \in S$ and $u_S(b)$ is defined. Since $u_S(b)$ is a twin of b and G is native, if $c = u_S(b)$ we have $@_c G \in S$ by Lemma 7. By construction of $\mathcal{M}^*$, aRc. By the inductive hypothesis, $\mathcal{M}_c^* \models G$, hence $\mathcal{M}_a^* \models \Diamond G$.
7. F is $\Diamond^- G$. Observe that, differently from the above case, G is necessarily native, because $@_a\Diamond^- c$ is not a relational formula (Lemma 1). By Remark B, since $a \in W$, there is a nominal b such that $@_b\Diamond a, @_b G \in S$ and $u_S(b)$ is defined. Since $u_S(b)$ is a twin of b and G is native, if $c = u_S(b)$ we have $@_c G \in S$ by Lemma 7. By the inductive hypothesis (since $c \in W$), we get $\mathcal{M}_c^* \models G$.

   Now, since $@_b\Diamond a \in S$, $c = u_S(b)$ and $a = u_S(a)$ (by remark A), by construction cRa. Therefore $\mathcal{M}_a^* \models \Diamond^- G$.

The following result lifts the model existence property upwards to the root set.

Lemma 9 *If Θ is a complete and open branch of a tableau rooted at S_0, then S_0 is satisfiable.*

Proof. We define an equivalence relation on nominals (with respect to the branch Θ) as follows: $a \sim b$ if $@_a b$ occurs in Θ. The relation $\approx$ is the reflexive, symmetric and transitive closure of $\sim$.

Now, let Θ be the sequence of nodes $S_0, ..., S_k$, where S_k is its leaf (hence a saturated and open node). Let $\mathcal{M}^* = \langle W, R, N^*, I\rangle$ be the model of S_k given by Lemma 8. Since N^* is undefined for nominals that do not occur in S_k, we can safely extend it to interpret all the nominals occurring in Θ. Let a_0 be any root nominal occurring in S_k. Then N is the extension of N^* such that for all nominals c occurring in Θ:

$$N(c) = \begin{cases} N^*(c) & \text{if } c \in W, \text{ i.e. } c \text{ occurs in } S_k \\ N^*(d) & \text{if for some } d \in W,\ c \approx d \\ a_0 & \text{otherwise} \end{cases}$$

It is clear that if $@_a b$ occurs in Θ, then $N(a) = N(b)$, and if c is deleted in Θ, then $N(c) = a_0$.

If $\mathcal{M} = \langle W, R, N, I\rangle$, obviously, it still holds that for every $@_a F \in S_k$, if $a \in W$ and F has not the form $\Diamond b$ for some b with $u_{S_k}(b)$ undefined, then $\mathcal{M}_{N(a)} \models F$.

We now prove that the satisfaction property propagates upwards, restricting our attention to nominals that are not deleted in Θ. Let us say that a formula $@_a F$ is *relevant* (w.r.t. Θ) iff either F is native, or both the following conditions hold:

- $@_a F$ contains only nominals that are never deleted in Θ, and
- F has not the form $\Diamond b$ for some b with $u_{S_k}(b)$ undefined;

Let us say that $\mathcal{M}$ is a Θ-model of a node S of Θ if for every relevant formula $@_a F \in S$, $\mathcal{M}_{N(a)} \models F$. Then we show that, for every $i = 0, ..., k-1$:

(•) if $\mathcal{M}$ is a Θ-model of S_{i+1}, then $\mathcal{M}$ is a Θ-model of S_i.

When $i = 0$ this is what we want, because the root set obviously contains only native (hence relevant) formulae.

In order to prove (•), the cases where S_{i+1} is obtained from S_i by applying a logical rule are trivial, since $S_i \subseteq S_{i+1}$.

So the only non-trivial case is the substitution rule, where:

$$\begin{aligned} S_i = &\ @_a b, S' \\ S_{i+1} = &\ S'^{\#}[a \mapsto b] \end{aligned}$$

By the induction hypothesis, for every relevant formula $@_c F \in S_{i+1} = S'^{\#}[a \mapsto b]$ $\mathcal{M}_{N(c)} \models F$. Note that all the descendants of a are deleted in Θ. Since $N(a) = N(b)$ (by definition), $\mathcal{M}_{N(a)} \models b$ and $\mathcal{M}_{N(a)} \models @_a b$.

Let now $@_c F$ be any relevant formula in $S' = S_i \backslash \{@_a b\}$ such that $@_c F \neq (@_c F)[a \mapsto b]$. If $@_c F$ is relevant then also $(@_c F)[a \mapsto b]$ is relevant. In fact:

- if $@_c F$ is native, then also $(@_c F)[a \mapsto b]$ is native because b is a root nominal (Lemma 1).
- If $@_c F$ contains only nominals that are never deleted in Θ, then the same holds for $(@_c F)[a \mapsto b]$. In fact, the only nominal possibly occurring in $(@_c F)[a \mapsto b]$ and not in $@_c F$ is b, and $b \in C_T$ (Lemma 1), so it cannot be deleted.
- If $@_c F$ is not a relational formula, obviously $(@_c F)[a \mapsto b]$ is not a relational formula either. So let us assume that $F = \Diamond d$ where $u_{S_k}(d)$ is defined. If $d \neq a$ there is nothing to prove ($(@_c \Diamond d)[a \mapsto b] = @_{c[a \mapsto b]} \Diamond d$ and $u_{S_k}(d)$ is defined).

  If $d = a$, then $(@_c \Diamond d)[a \mapsto b] = @_{c[a \mapsto b]} \Diamond b$. Since $\Diamond b$ is native, $(@_c F)[a \mapsto b]$ is relevant.

Therefore by the inductive hypothesis

$$\mathcal{M}_{N(c[a \mapsto b])} \models F[a \mapsto b]$$

where $c[a \mapsto b] = b$ if $c = a$, and $c[a \mapsto b] = c$ otherwise. Since $N(a) = N(b)$, $\mathcal{M}_{N(c[a \mapsto b])} \models F$. If $c[a \mapsto b] = c$, we are done. Otherwise, if $c[a \mapsto b] = b$ then $c = a$, so $N(c[a \mapsto b]) = N(b) = N(a) = N(c)$. Hence, also in this case $\mathcal{M}_{N(c)} \models F$.

Theorem 2 (Completeness) *If S is unsatisfiable, then every complete tableau for S is closed.*

Proof. Completeness follows directly from Lemma 9 and the fact that nominals occurring in the initial set are never deleted.

References

[1] T. Bolander and P. Blackburn. Terminating tableau calculi for hybrid logics extending K. In S. Demri and C. Areces, editors, *Methods for Modalities 5*, Cachan, France, 2007.

[2] T. Bolander and P. Blackburn. Termination for hybrid tableaus. *Journal of Logic and Computation*, 2007.

[3] S. Cerrito and M. Cialdea Mayer. An efficient approach to nominal equalities in hybrid logic tableaux. *Journal of Applied Non-classical Logics*. To appear.

[4] S. Cerrito and M. Cialdea Mayer. Terminating tableaux for HL(@) without loop-checking. Technical Report IBISC-RR-2007-07, Ibisc Lab., Université d'Evry Val d'Essonne, 2007. Available at `http://www.ibisc.univ-evry.fr/Vie/TR/2007/IBISC-RR2007-07.pdf`.

[5] M. Cialdea Mayer, S. Cerrito, E. Benassi, F. Giammarinaro, and C. Varani. Two tableau provers for basic hybrid logic. Technical Report RT-DIA-145-2009, Dipartimento di Informatica e Automazione, Università di Roma Tre, 2009. Available at `http://cialdea.dia.uniroma3.it/herod/`.