Dipartimento di Informatica e Automazione
Via della Vasca Navale, 79
00146 Roma, Italy

A calculus for a decidable fragment of hybrid logic with binders

Serenella Cerrito[1], Marta Cialdea Mayer[2]

RT-DIA-181-2011 January 2011

[1] Lab. Ibisc, Université d'Evry Val d'Essonne, France
[2] Università di Roma Tre, Italy

ABSTRACT

In this paper we provide the first (as far as we know) direct calculus deciding satisfiability of formulae in negation normal form in the fragment of hybrid logic with the satisfaction operator and the binder, where no binder dominates any occurrence of the $\Box$ operator.

A preliminary transformation of formulae into equisatisfiable ones turns the calculus into a satisfiability decision procedure for formulae in $\mathsf{HL}(@,\downarrow)\setminus\Box\downarrow\Box$, *i.e.* formulae in negation normal form where no occurrence of the binder is both in the scope of and dominates a $\Box$ operator.

The calculus is based on tableaux, where nominal equalities are treated by means of substitution, and termination is achieved by means of a form of anywhere blocking with indirect blocking. Direct blocking is a relation between nodes in a tableau branch, holding whenever the respective labels (formulae) are equal up to nominal renaming. Indirect blocking is based on a partial order on the nodes of a tableau branch, which arranges them into a tree-like structure.

Contents

1 Introduction

The Hybrid Logic $\mathsf{HL}(@, \downarrow)$ is an extension of modal (propositional, possibly multi-modal) logic K by means of three constructs: *nominals* (propositions which hold in exactly one state of the model), the *satisfaction operator* @ (allowing one to state that a given formula holds at the state named by a given nominal), and the *binder* $\downarrow$, accompanied by *world variables*, which allows one to give a name to the current state (see [2] for an overview of the subject).

The satisfiability problem for formulae of basic Hybrid Logic $\mathsf{HL}(@)$ (without the binder) is decidable,[1] and it stays decidable even with the addition of other operators, such as the global and converse modalities. On the contrary, an unrestricted addition of the binder causes a loss of decidability [1, 3].[2]

However, similarly to what happens for first order logic, one can obtain decidable fragments of hybrid logic with the binder by imposing syntactic restrictions on the way formulae are built. Some decidability results are proved in [12], which considers full Hybrid Logic $\mathsf{HL}(@, \downarrow, \mathsf{E}, \Diamond^-)$ – that will henceforth be abbreviated as FHL. In such a work it is proved that the source of undecidability is the occurrence of a specific *modal pattern* in formulae in negation normal form (NNF). A pattern π is a sequence of operators, and a formula is a π-formula, where $\pi = Op_1...Op_n$, if it is in NNF and contains some occurrence of Op_1 which dominates an occurrence of Op_2, which in turn dominates an occurrence of Op_3, etc. For simplicity, moreover, when the $\Box$ operator is used in a pattern, it actually stands for any *universal operator*, *i.e.* one of the modalities $\Box, \Box^-$ or A. In particular, a $\Box \downarrow$-formula is a hybrid formula in NNF where some occurrence of the binder is in the scope of a universal operator; a $\downarrow \Box$-formula is a hybrid formula in NNF where some occurrence of the universal operator is in the scope of a binder; and a $\Box \downarrow \Box$-formula is a hybrid formula in NNF containing a universal operator which dominates a binder, which in turn dominates a universal operator. Finally, if π is a pattern, the fragment $\mathsf{HL}(Op_1, ..., Op_k) \setminus \pi$ is constituted by the class of NNF hybrid formulae in $\mathsf{HL}(Op_1, ..., Op_k)$ excluding π-formulae.

An important decidability result proved in [12] is the following:

1. The satisfiability problem for $\mathsf{FHL} \setminus \Box \downarrow \Box$ is decidable.

Such a result is tight, in the sense that there is no pattern π that contains $\Box \downarrow \Box$ as a subsequence and such that the satisfiability problem for $\mathsf{FHL} \setminus \pi$ is still decidable. Therefore, the fragment $\mathsf{FHL} \setminus \Box \downarrow \Box$ is particularly interesting.

For the aim of the present work, it is important to recall the intermediate results allowing [12] to prove 1:

2. The satisfiability problem for $\mathsf{FHL} \setminus \Box \downarrow$ is decidable. This is proved by showing that there exists a satisfiability preserving translation from $\mathsf{FHL} \setminus \Box \downarrow$ to $\mathsf{HL}(@, \mathsf{E}, \Diamond^-)$. The translation is obtained by first replacing any occurrence of the binder by a full existential quantification over worlds (*i.e.* $\downarrow x.F$ is replaced by $\exists x(x \wedge F)$), then transforming the resulting formula into prenex normal form (which can be

[1] The notation $\mathsf{HL}(Op_1, ..., Op_n)$ is commonly used to denote the extension of modal logic K by means of the operators $Op_1...Op_n$. In particular, $\mathsf{HL}(@, \downarrow, \mathsf{E}, \Diamond^-)$ and $\mathsf{HL}(@, \mathsf{E}, \Diamond^-)$ include the existential global modality E (and its dual A) and the converse operator $\Diamond^-$ (and its dual $\Box^-$).

[2] The cited works prove a stronger result: even in the absence of nominals and @, $\mathsf{HL}(\downarrow)$ is undecidable.

done because no universal operator dominates a binder), and skolemizing away the existential operators by use of fresh nominals.

3. The satisfiability problem for $\mathsf{FHL} \setminus \downarrow \Box$ is decidable. This holds because the standard translation ST of FHL into first order classical logic [1, 12] maps formulae in the considered fragment into *universally guarded formulae* [12], that have a decidable satisfiability problem [7].

Result 1 easily follows from 2 and 3. Let in fact F be any formula in $\mathsf{FHL} \setminus \Box \downarrow \Box$. Any occurrence of the binder dominating a universal operator is not, in its turn, dominated by a universal operator. Therefore it can be skolemized away like in the proof of 2. Repeating this transformation for every $\downarrow \Box$-subformula of F, a formula F' is obtained, where no occurrence of the binder dominates a universal operator and which is satisfiable if and only if F is satisfiable. Satisfiability of F' can be decided because of result 3.

The above sketched approach to proving result 1 shows also that any decision procedure for formulae in $\mathsf{FHL} \setminus \downarrow \Box$ can easily be turned into a decision procedure for formulae in the largest fragment $\mathsf{FHL} \setminus \Box \downarrow \Box$, by means of a preliminary formula rewriting.

In particular, satisfiability of formulae in the fragment $\mathsf{FHL} \setminus \Box \downarrow \Box$ can be tested by translation, by use of any calculus for the guarded fragment, such as the tableau calculi defined in [8, 9], or the decision procedure based on resolution given in [6]. However, the translation of an object formula F into a guarded formula is quite cumbersome. In fact, the standard translation of F is a *universally guarded* formula, which has to be rewritten into an equisatisfiable *guarded* one [7]. Moreover, decision procedures such as the above mentioned ones apply to constant-free formulae. Since formulae obtained from the translation may in general contain constants (coming from nominals), a further rewriting would be necessary to eliminate them (by introducing new predicates) [7, 11].

Beyond the generally recognized interest of having direct calculi for modal logics, we therefore consider that the problem of defining direct decision procedures for decidable fragments of hybrid logics deserves a specific attention.

In this paper we provide the first (as far as we know) direct calculus deciding satisfiability of formulae in $\mathsf{HL}(@, \downarrow) \setminus \downarrow \Box$. A preliminary transformation of formulae into equisatisfiable ones, like explained above, turns the calculus into a decision procedure for satisfiability of formulae in $\mathsf{HL}(@, \downarrow) \setminus \Box \downarrow \Box$.

The work is organized as follows. In Section 2 we recall the syntax and semantics of $\mathsf{HL}(@, \downarrow)$. In Section 3 we define the tableau system, and in Section 4 we prove its termination. Soundness being trivial, we devote Section 5 to prove that the system is complete with respect to unsatisfiability. Section 6 concludes this work, and includes a comparison of some aspects of our work with techniques already present in the literature.

2 Hybrid Logic

Let PROP (the set of propositional letters) and NOM (the set of nominals) be disjoint sets of symbols. Let VAR be a set of *world variables*. Hybrid formulae F in $\mathsf{HL}(@, \downarrow)$ are defined by the following grammar:

$$F := \ p \ | \ a \ | \ x \ | \ \neg F \ | \ F \wedge F \ | \ F \vee F \ | \ \Diamond F \ | \ \Box F \ | \ t : F \ | \ \downarrow x.F$$

where $p \in \mathsf{PROP}$, $a \in \mathsf{NOM}$, $x \in \mathsf{VAR}$ and $t \in \mathsf{VAR} \cup \mathsf{NOM}$. In this work, the notation $t : F$ is used rather than the more usual one $@_t F$. We use metavariables a, b, c, possibly with subscripts, for nominals, while x, y, z are used for variables.

A formula of the form $a : F$ is called a *satisfaction statement*, whose *outermost nominal* is a, F is its *body*, and the satisfaction symbol applied to a and F is the *outermost satisfaction symbol* of the statement. The operator $\downarrow$ is a *binder* for (world) variables. A variable x is *free* in a formula if it does not occur in the scope of a $\downarrow x$. A formula is *ground* if it contains no free variables.

A *subformula* of a formula F is a substring of F (possibly F itself) that is itself a formula. An *instance* of a formula F is an expression obtained by replacing every free variable of F with some nominal. A subformula may contain free variables, while an instance is always a ground formula. Obviously, if a subformula F of G does not contain any free variable, then F is both a subformula of G and an instance of a subformula of G. For instance $x : p$ is a subformula of $\downarrow x.x : p$, but is not an instance of any formula, $a : p$ is an instance of a subformula of $\downarrow x.x : p$ but it is not a subformula of $\downarrow x.x : p$, while p is both a subformula of $\downarrow x.x : p$. and an instance of a subformula of such a formula.

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

A *variable assignment* σ for $\mathcal{M}$ is a function $\mathsf{VAR} \rightarrow W$. If $x \in \mathsf{VAR}$ and $w \in W$, the notation σ_w^x stands for the variable assignment σ' such that: $\sigma'(y) = \sigma(y)$ if $y \neq x$ and $\sigma'(x) = w$.

If $\mathcal{M} = \langle W, R, N, I \rangle$ is an interpretation, $w \in W$, σ is a variable assignment for $\mathcal{M}$ and F is a formula, the relation $\mathcal{M}_w, \sigma \models F$ is inductively defined as follows:

1. $\mathcal{M}_w, \sigma \models p$ if $p \in I(w)$, for $p \in \mathsf{PROP}$.
2. $\mathcal{M}_w, \sigma \models a$ if $N(a) = w$, for $a \in \mathsf{NOM}$.
3. $\mathcal{M}_w, \sigma \models x$ if $\sigma(x) = w$, for $x \in \mathsf{VAR}$.
4. $\mathcal{M}_w, \sigma \models \neg F$ if $\mathcal{M}_w, \sigma \not\models F$.
5. $\mathcal{M}_w, \sigma \models F \wedge G$ if $\mathcal{M}_w, \sigma \models F$ and $\mathcal{M}_w, \sigma \models G$.
6. $\mathcal{M}_w, \sigma \models F \vee G$ if either $\mathcal{M}_w, \sigma \models F$ or $\mathcal{M}_w, \sigma \models G$.
7. $\mathcal{M}_w, \sigma \models a : F$ if $\mathcal{M}_{N(a)}, \sigma \models F$, for $a \in \mathsf{NOM}$.
8. $\mathcal{M}_w, \sigma \models x : F$ if $\mathcal{M}_{\sigma(x)}, \sigma \models F$, for $x \in \mathsf{VAR}$.
9. $\mathcal{M}_w, \sigma \models \Box F$ if for each w' such that wRw', $\mathcal{M}_{w'}, \sigma \models F$.
10. $\mathcal{M}_w, \sigma \models \Diamond F$ if there exists w' such that wRw' and $\mathcal{M}_{w'}, \sigma \models F$.
11. $\mathcal{M}_w, \sigma \models \downarrow x.F$ if $\mathcal{M}_w, \sigma_w^x \models F$.

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

$\mathcal{M}_w, \sigma \models F$ if and only if $\mathcal{M}_w, \sigma \models G$. A formula F holds in a state w of a model $\mathcal{M}$ ($\mathcal{M}_w \models F$) iff $\mathcal{M}_w, \sigma \models F$ for every variable assignment σ.

It is worth pointing out that, if $t \in \mathsf{VAR} \cup \mathsf{NOM}$ and F is a formula:

$$\neg(t : F) \equiv t : \neg F \qquad \neg \downarrow x.F \equiv \downarrow x.\neg F \qquad \neg\Diamond F \equiv \Box\neg F \qquad \neg\Box F \equiv \Diamond\neg F$$

This allows us to restrict our attention to formulae in negation normal form (where negation dominates only atoms), in the sequel.

3 The Tableau Calculus

3.1 The Expansion Rules

A *tableau branch* is a sequence of *nodes* $n_0, n_1, \ldots$, where each node is labelled by a ground satisfaction statement, and a tableau is a set of branches. If n occurs before m in the branch $\mathcal{B}$, we shall write $n < m$. The label of the node n will be denoted by $label(n)$. The notation $(n)\, a : F$ will be used to denote the node n, and simultaneously say that its label is $a : F$.

A tableau for a formula F is initialized with a single branch, constituted by the single node $(n_0)\, a_0 : F$, where a_0 is a new nominal. The formula $a_0 : F$ is the *initial formula* of the tableau, which is assumed to be ground.

A tableau is expanded by application of the rules in Table 1, which are applied to a given branch. Their reading is standard: a rule is applicable if the branch contains a node (two nodes) labelled by the formula(e) shown as premiss(es) of the rules. The rules $\wedge, @, \downarrow, \Box$ and $\Diamond$ add one or two nodes to the branch, labelled by the conclusion(s); the rule $\vee$ replaces the current branch $\mathcal{B}$ with two branches, each of which is obtained by adding $\mathcal{B}$ a new node, labelled, respectively, by the formulae shown on the left and right below the inference line.

The $\Box$ rule has two premisses, which must both occur in the branch, in any order. The leftmost premiss of the $\Box$ rule is called its *major premiss*, the rightmost one its *minor premiss*. The minor premiss is a *relational formula*, *i.e.* a satisfaction statement of the form $a : \Diamond b$ (where b is a nominal). A formula of the form $\Box F$ is called a *universal formula*.

The $\Diamond$ rule is called *blockable rule*, a formula of the form $a : \Diamond F$, where F is not a nominal, is a *blockable formula* and a node labelled by a blockable formula is a *blockable node*.

If F is a formula, the notation $F[a/x]$ is used to denote the formula that is obtained from F replacing a for every free occurrence of the variable x. Analogously, if a and b are nominals, $F[b/a]$ is the formula obtained from F replacing b for every occurrence of a. The *equality rule* ($=$) does not add any node to the branch, but modifies the labels of its nodes. The schematic formulation of such a rule in Table 1 indicates that it can be fired whenever a branch $\mathcal{B}$ contains a *nominal equality* of the form $a : b$ (with $a \neq b$); as a result of the application of the rule, every node label F in $\mathcal{B}$ is replaced by $F[b/a]$.

The first node of a branch $\mathcal{B}$ is called the *top node* and its label the *top formula* of $\mathcal{B}$. The nominals occurring in the top formula are called *top nominals*. If the top node of $\mathcal{B}$ is n_0, the branch is said to be *rooted* at n_0. Note that the notion of top nominal is relative to a tableau branch. In fact, applications of the equality rule may change the top formula, hence the set of top nominals.

$$\frac{a:(F\wedge G)}{\begin{array}{c}a:F\\ a:G\end{array}}\ (\wedge) \qquad \frac{a:(F\vee G)}{a:F \quad \mid \quad a:G}\ (\vee)$$

$$\frac{a:b:F}{b:F}\ (@) \qquad \frac{a:\downarrow x.F}{a:F[a/x]}\ (\downarrow) \qquad \frac{\begin{array}{c}[\mathcal{B}]\\ a:b\end{array}}{\mathcal{B}[b/a]}\ (=)$$

(not applicable if $a=b$)

$$\frac{a:\Box F \quad a:\Diamond b}{b:F}\ (\Box) \qquad \frac{a:\Diamond F}{\begin{array}{c}a:\Diamond b\\ b:F\end{array}}\ (\Diamond)$$

where b is a new nominal
(not applicable if F is a nominal)

Table 1: Expansion rules

It is worth noticing that, since we assume that the initial formula of a tableau is ground, node labels in any branch are always ground formulae.

In the following definition, the current branch is left implicit, so as to lighten the notation.

Definition 1. *If a node* n *is added to a branch by application of the rule* $\mathcal{R}$ *to the node* m *then we write* $m \leadsto^{\mathcal{R}} n$*. In the case of rules with two conclusions, we write* $m \leadsto^{\mathcal{R}} (n,k)$*, or, sometimes,* $m \leadsto^{\mathcal{R}} n$ *and* $m \leadsto^{\mathcal{R}} k$*. In the case of the two premisses rule* $\Box$ *we write* $(m,k) \leadsto^{\Box} n$*.*

Note that the application of the equality rule does not change nodes, but only their labels, therefore it does not change the relation $\leadsto^{\mathcal{R}}$.

We say that a formula $a:F$ occurs in a tableau branch $\mathcal{B}$ (or $a:F\in\mathcal{B}$) if for some node n of the branch, $\mathit{label}(n) = a:F$. Similarly, a nominal occurs in a branch $\mathcal{B}$ if it occurs in the label of some node of $\mathcal{B}$. Finally, a nominal a *labels* a formula F in $\mathcal{B}$ if $a:F\in\mathcal{B}$.

3.2 Restrictions on Rule Application

Termination is achieved by means of a loop-checking mechanism, that requires some preliminary definitions.

Definition 2 (Nominal compatibility)**.** *If* $\mathcal{B}$ *is a tableau branch and* a *is a nominal occurring in* $\mathcal{B}$*, then*

$$\Phi_{\mathcal{B}}(a) = \{p \mid p\in \mathsf{PROP} \text{ and } a:p\in\mathcal{B}\} \cup \{\Box F \mid a:\Box F\in\mathcal{B}\}$$

If a *and* b *are nominals occurring in a tableau branch* $\mathcal{B}$*, then* a *and* b *are* compatible *in* $\mathcal{B}$ *if* $\Phi_{\mathcal{B}}(a)=\Phi_{\mathcal{B}}(b)$*,* i.e. *if they label the same propositions in* PROP *and the same universal formulae.*

Definition 3 (Mappings and matching)**.** *A mapping* π *for a branch* $\mathcal{B}$ *is an injective function from* non-top *nominals to* non-top *nominals such that for all* a*,* a *and* $\pi(a)$ *are compatible in* $\mathcal{B}$*.*

A mapping π *for* $\mathcal{B}$ *matches a formula* F *to a formula* G *if:*

1. $\pi(F) = G$*;*

2. π *is the identity for all nominals which do not occur in* F*.*

A formula F *can be matched to a formula* G *in* $\mathcal{B}$ *if there exists a mapping* π *for* $\mathcal{B}$ *matching* F *to* G*.*

Note that, since mappings are injective, if F can be mapped to G then G can be mapped to F, too.

Since a mapping π is the identity almost everywhere, it can be represented by a finite set of pairs of the form $\{b_1/a_1, ..., b_n/a_n\}$ where $a_i \neq b_i$, whenever $\pi(a_i) = b_i$ and $\pi(c) = c$ for all $c \notin \{a_1, ..., a_n\}$.

The application of the blockable rule is restricted by blocking conditions: a direct blocking condition, which forbids the application of the blockable rule to a node n, whenever the label of a previous node can be matched to $label(n)$; and also an indirect blocking condition. In fact, since a node may be (directly) blocked in a branch after that it has already been expanded, all the nodes which, in some sense, depend from such an expansion must be blocked too. So, a notion of indirect blocking is needed, which in turn requires a new partial order on nodes.

The following definition introduces a binary relation on nodes, which organizes them into a family of trees.

Definition 4. *Let* $\mathcal{B}$ *be a tableau branch. The relation* $n \prec_{\mathcal{B}} m$ *between nodes of* $\mathcal{B}$ *is inductively defined as follows:*

Base case *If* $n \rightsquigarrow^{\Diamond} (m, k)$*, then* $n \prec_{\mathcal{B}} m$ *and* $n \prec_{\mathcal{B}} k$*;*

Inductive cases *If* $m \prec_{\mathcal{B}} n$*, then:*

1. *if* $n \rightsquigarrow^{\mathcal{R}} k$*, where* $\mathcal{R} \in \{\vee, @, \downarrow, \wedge\}$*, then* $m \prec_{\mathcal{B}} k$*;*
2. *if* $label(n)$ *is a relational formula and for some* n'*,* $(n', n) \rightsquigarrow^{\Box} k$*, then* $m \prec_{\mathcal{B}} k$*.*

If $m \prec_{\mathcal{B}} n$ *then* n *is said to be a* child *of* m w.r.t. $\prec_{\mathcal{B}}$*, and* m *the* parent *of* n*. A node* n *in* $\mathcal{B}$ *is called a* root node *if it has no parent. Two nodes* n *and* k *are called* siblings *if either both of them are root nodes, or for some* m*,* $m \prec_{\mathcal{B}} n$ *and* $m \prec_{\mathcal{B}} k$*.*

The relation $\prec^+_{\mathcal{B}}$ *is the transitive closure of* $\prec_{\mathcal{B}}$*. If* $n \prec^+_{\mathcal{B}} m$*, then* n *is an* ancestor *of* m *and* m *a descendant of* n w.r.t. $\prec_{\mathcal{B}}$*.*

In other terms, when the blockable rule is applied to a node n, a first pair of children of n *w.r.t.* $\prec_{\mathcal{B}}$ is generated. The application of rules other than $\Diamond$ generates siblings, where, in the case of the two premisses rule $\Box$, it is the minor premiss which is added a sibling. Intuitively, when $n \prec_{\mathcal{B}} m$, n is the node which is taken to be the main "responsible" of the presence of m in the branch. In fact, the first "children" of a node n are nodes obtained from n by application of the blockable rule. And, if a node m is obtained from m' (as the minor premiss, in the case of the $\Box$ rule) by means of applications of non-blockable rules, then they are "siblings" *w.r.t.* $\prec_{\mathcal{B}}$.

Example 1. *As an example, consider the tableau branch for*

$$F = a : (\Diamond p \wedge \Box \downarrow x.\Diamond(p \wedge \neg x \wedge \downarrow y.a : \Diamond y))$$

represented in Figure 1. Node numbering reflects the order in which nodes are added to the branch. The right column reports the $\rightsquigarrow^{\mathcal{R}}$ *relation justifying the addition of the corresponding node to the branch.* W.r.t. *the relation* $\prec_{\mathcal{B}}$*,* $0, 1$ *and* 3 *are root nodes with no children;* 2 *is also a root node, with children* $4, 5, 6$ *and* 7*; nodes* 8–17 *are all children of* 7*.*

(0) $a_0 : a : (\Diamond p \wedge \Box \downarrow x.\Diamond(p \wedge \neg x \wedge \downarrow y.a : \Diamond y))$	
(1) $a : (\Diamond p \wedge \Box \downarrow x.\Diamond(p \wedge \neg x \wedge \downarrow y.a : \Diamond y))$	$0 \rightsquigarrow^{@} 1$
(2) $a : \Diamond p$	$1 \rightsquigarrow^{\wedge} 2$
(3) $a : \Box \downarrow x.\Diamond(p \wedge \neg x \wedge \downarrow y.a : \Diamond y)$	$1 \rightsquigarrow^{\wedge} 3$
(4) $a : \Diamond b$	$2 \rightsquigarrow^{\Diamond} 4$
(5) $b : p$	$2 \rightsquigarrow^{\Diamond} 5$
(6) $b :\downarrow x.\Diamond(p \wedge \neg x \wedge \downarrow y.a : \Diamond y)$	$(3, 4) \rightsquigarrow^{\Box} 6$
(7) $b : \Diamond(p \wedge \neg b \wedge \downarrow y.a : \Diamond y)$	$6 \rightsquigarrow^{\downarrow} 7$
(8) $b : \Diamond c$	$7 \rightsquigarrow^{\Diamond} 8$
(9) $c : p \wedge \neg b \wedge \downarrow y.a : \Diamond y$	$7 \rightsquigarrow^{\Diamond} 9$
(10) $c : p \wedge \neg b$	$9 \rightsquigarrow^{\wedge} 10$
(11) $c :\downarrow y.a : \Diamond y$	$9 \rightsquigarrow^{\wedge} 11$
(12) $c : p$	$10 \rightsquigarrow^{\wedge} 12$
(13) $c : \neg b$	$10 \rightsquigarrow^{\wedge} 13$
(14) $c : a : \Diamond c$	$11 \rightsquigarrow^{\downarrow} 14$
(15) $a : \Diamond c$	$14 \rightsquigarrow^{@} 15$
(16) $c :\downarrow x.\Diamond(p \wedge \neg x \wedge \downarrow y.a : \Diamond y)$	$(3, 15) \rightsquigarrow^{\Box} 16$
(17) $c : \Diamond(p \wedge \neg c \wedge \downarrow y.a : \Diamond y)$	$16 \rightsquigarrow^{\downarrow} 17$

Figure 1: A tableau branch for $a : (\Diamond p \wedge \Box \downarrow x.\Diamond(p \wedge \neg x \wedge \downarrow y.a : \Diamond y))$

Before defining the blocking conditions we prove some important properties of $\prec_{\mathcal{B}}$: *i.e.* that any node has at most one parent and only blockable nodes may have children. Consequently, $\prec_{\mathcal{B}}$ arranges the nodes of a branch into a forest of trees, where non-terminal nodes are blockable nodes. For instance, the nodes of the tableau branch $\mathcal{B}$ of Example 1 are arranged into four trees: three of them consist of a single node (respectively: 0, 1 and 3), while the fourth one is rooted at 2 and is shown in Figure 2.

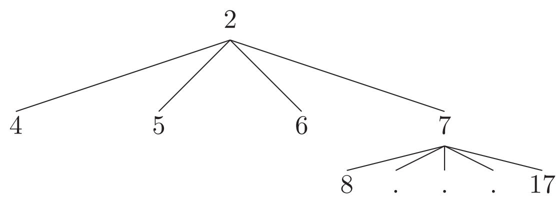

Figure 2: One of the trees induced by the $\prec_{\mathcal{B}}$ relation on the nodes in the branch of Figure 1

Lemma 1.

1. *For each node n in a tableau branch $\mathcal{B}$, there exists at most one node m such that $m \prec_{\mathcal{B}} n$. Therefore, there is exactly one maximal chain*

$$n_1 \prec_{\mathcal{B}} n_2 \prec_{\mathcal{B}} ... \prec_{\mathcal{B}} n_k = n$$

where n_1 is any root node.

2. *If for some n, $m \prec_{\mathcal{B}} n$, then m is a blockable node. Therefore, for any chain*

$$n_1 \prec_{\mathcal{B}} n_2 \prec_{\mathcal{B}} ... \prec_{\mathcal{B}} n_k \prec_{\mathcal{B}} n_{k+1}$$

$n_1, ..., n_k$ are all blockable nodes.

Proof. The first item can easily be proved by induction on the definition of $\prec_{\mathcal{B}}$. The second follows directly from the definition. □

As a final remark on the relation $\prec_{\mathcal{B}}$, it is worth pointing out that two siblings are not necessarily labelled by satisfaction statements whose outermost nominal is the same (because of the @ rule), and, *vice-versa*, not all nodes labelled by formulae of the form $b : F$ for a given nominal b are necessarily siblings. For instance, let us assume that, in a given branch $\mathcal{B}$, $n_1 \prec_{\mathcal{B}} (m)\, a : F$ and $n_2 \prec_{\mathcal{B}} (k)\, b : G$; if then the equality rule replaces a with b, we still have, in the new branch $n_1 \prec_{\mathcal{B}} (m)\, b : F$ and $n_2 \prec_{\mathcal{B}} (k)\, b : G$. So, if $n_1 \neq n_2$, m and k are not siblings, although the outermost nominal of their label is the same.

We can now define the notions of direct and indirect blocking.

Definition 5 (Direct and indirect blocking)**.** *A node $(n)\, a : \Diamond F$ is* directly blocked *by $(m)\, b : \Diamond G$ in $\mathcal{B}$ if*

- *$m < n$, m is neither directly blocked in $\mathcal{B}$ nor it has any ancestor* w.r.t. *$\prec_{\mathcal{B}}$ which is directly blocked in $\mathcal{B}$;*
- *$b : G$ can be matched to $a : F$ in $\mathcal{B}$.*

A node n is directly blocked in $\mathcal{B}$ if it is blocked by some m in $\mathcal{B}$, and it is indirectly blocked *in $\mathcal{B}$ if it has an ancestor* w.r.t. *$\prec_{\mathcal{B}}$ which is directly blocked in $\mathcal{B}$. An indirectly blocked node is called a* phantom node *(or, simply, a phantom).*

The tableau branch $\mathcal{B}$ represented in Figure 1 represents a blocking case: node 17 is blocked by 7, because b e c are compatible ($\Phi_{\mathcal{B}}(b) = \Phi_{\mathcal{B}}(c) = \{p\}$).

It must be remarked that the blocking relation is dynamic, *i.e.* blockings are not established forever, since they are relative to a tableau branch, and can be undone when expanding the branch. In fact, a node may be blocked in a branch $\mathcal{B}$ and then unblocked after expanding $\mathcal{B}$, because the addition of new nodes or changes in node labels may destroy nominal compatibility. Similarly, when the equality rule affects either the label of a blocked node n or that of its blocking node, n is not automatically kept blocked. Possibly, a new blocking can be introduced (but compatibilities must be checked again), by means of a different mapping.

The application of the expansion rules is restricted by the following conditions:

Definition 6 (Restrictions on the expansion rules)**.** *The expansion of a tableau branch $\mathcal{B}$ is subject to the following restrictions:*

R1. *no node labelled by a formula already occurring in $\mathcal{B}$ as the label of a* non-phantom *node is ever added to $\mathcal{B}$;*

R2. *a node n cannot be expanded if it is a blockable node and there are $k_0, k_1 \in \mathcal{B}$ such that $n \rightsquigarrow^{\Diamond} (k_0, k_1)$;*

R3. *a phantom node cannot be expanded by means of a single-premiss rule, nor can it be used as the minor premiss of the $\Box$ rule;*

R4. *a node n cannot be expanded if it is a blockable node and n is directly blocked in $\mathcal{B}$.*

Restriction **R1** amounts to saying that:

1. a node n (or pair of nodes n, m) cannot be expanded in $\mathcal{B}$ if the expansion of n (and m) would produce a single node, whose label would be a formula which already occurs in $\mathcal{B}$ as the label of a non-phantom node;
2. the $\wedge$ rule cannot be applied to a node $(n)\, a : F \wedge G$ whenever both $a : F$ and $a : G$ are already the labels of non-phantom nodes in $\mathcal{B}$;
3. if a node $(n)\, a : F \wedge G$ can be expanded, but $a : F$ (or $a : G$, but not both) is already the label of a non-phantom node, then only one new node is added to the branch, with label $a : G$ (or $a : F$).

Note that restriction **R2** does not require k_0 and k_1 to be non-phantom (differently from **R1**). In fact, if $n \rightsquigarrow^{\Diamond} (k_0, k_1)$ and k_0, k_1 are phantom nodes, then n is either directly blocked or it is itself a phantom; in either case it cannot be expanded, by restrictions **R3** and **R4**. Finally, let us observe that restriction **R3** does not forbid firing the $\Box$ rule with a non-phantom minor premiss, even if the major premiss is a phantom node.

It is worth pointing out that termination would not be guaranteed if restriction **R1** were replaced by the condition that a node (or pair of nodes) is never expanded more than once on the branch. This is shown by Example 2.

Example 2. *The construction of the following tableau branch for $a : (\Diamond b \wedge \Box a : \Diamond b)$ satisfies the requirement that no node or pair of nodes are ever expanded more than once, but violates restriction* **R1***:*

(0)	$a_0 : a : (\Diamond b \wedge \Box(a : \Diamond b))$	
(1)	$a : (\Diamond b \wedge \Box(a : \Diamond b))$	$0 \rightsquigarrow^{@} 1$
(2)	$a : \Diamond b$	$1 \rightsquigarrow^{\wedge} 2$
(3)	$a : \Box(a : \Diamond b)$	$1 \rightsquigarrow^{\wedge} 3$
(4)	$b : a : \Diamond b$	$(3,2) \rightsquigarrow^{\Box} 4$
(5)	$a : \Diamond b$	$4 \rightsquigarrow^{@} 5$
(6)	$b : a : \Diamond b$	$(3,5) \rightsquigarrow^{\Box} 6$
(7)	$a : \Diamond b$	$6 \rightsquigarrow^{@} 7$
	....	

Obviously, tableau construction does not terminate.

A branch is *closed* whenever it contains, for some nominal a, either a pair of nodes $(n)\, a : p$, $(m)\, a : \neg p$ for some $p \in \mathsf{PROP}$, or a node $(n)\, a : \neg a$. As usual, we assume that a closed branch is never expanded further on. A branch which is not closed is *open*. A branch is *complete* when it cannot be further expanded. For instance, the tableau branch represented in Figure 1 is complete and open.

3.3 Examples

This section concludes with some further examples. In each of them, $\mathcal{B}$ denotes the considered branch, and the notation $\mathcal{B}_n$ is used to denote the branch segment up to node n, while Φ_n abbreviates $\Phi_{\mathcal{B}_n}$.

Example 3. *Figure 3 represents a closed one-branch tableau for*

$$F = (\Diamond \downarrow x.\Diamond(x : p)) \wedge (\Diamond \downarrow y.\Diamond(y : \neg p)) \wedge (\Diamond \downarrow z.(\Diamond(z : p) \wedge \Diamond(z : \neg p)))$$

where the first applications of the $\wedge$*-rule are collapsed into one.*

(0)	$a_0 : F$	
(1)	$a_0 : \Diamond \downarrow x.\Diamond x : p$	$0 \leadsto^{\wedge} 1$
(2)	$a_0 : \Diamond \downarrow y.\Diamond y : \neg p$	$0 \leadsto^{\wedge} 2$
(3)	$a_0 : \Diamond \downarrow z.(\Diamond(z : p) \wedge \Diamond(z : \neg p))$	$0 \leadsto^{\wedge} 3$
(4)	$a_0 : \Diamond a$	$1 \leadsto^{\Diamond} 4$
(5)	$a :\downarrow x.\Diamond x : p$	$1 \leadsto^{\Diamond} 5$
(6)	$a : \Diamond a : p$	$5 \leadsto^{\downarrow} 6$
(7)	$a : \Diamond a_1$	$6 \leadsto^{\Diamond} 7$
(8)	$a_1 : a : p$	$6 \leadsto^{\Diamond} 8$
(9)	$a : p$	$8 \leadsto^{@} 9$
(10)	$a_0 : \Diamond b$	$2 \leadsto^{\Diamond} 10$
(11)	$b :\downarrow y.\Diamond y : \neg p$	$2 \leadsto^{\Diamond} 11$
(12)	$b : \Diamond b : \neg p$	$11 \leadsto^{\downarrow} 12$
(13)	$b : \Diamond b_1$	$12 \leadsto^{\Diamond} 13$
(14)	$b_1 : b : \neg p$	$12 \leadsto^{\Diamond} 14$
(15)	$b : \neg p$	$14 \leadsto^{@} 15$
(16)	$a_0 : \Diamond c$	$3 \leadsto^{\Diamond} 16$
(17)	$c :\downarrow z.(\Diamond(z : p) \wedge \Diamond(z : \neg p))$	$3 \leadsto^{\Diamond} 17$
(18)	$c : \Diamond c : p \wedge \Diamond c : \neg p$	$17 \leadsto^{\downarrow} 18$
(19)	$c : \Diamond c : p$	$18 \leadsto^{\wedge} 19$
(20)	$c : \Diamond c : \neg p$	$18 \leadsto^{\wedge} 20$
(21)	$c : \Diamond c_1$	$19 \leadsto^{\Diamond} 21$
(22)	$c_1 : c : p$	$19 \leadsto^{\Diamond} 22$
(23)	$c : p$	$22 \leadsto^{@} 23$
(24)	$c : \Diamond c_2$	$20 \leadsto^{\Diamond} 24$
(25)	$c_2 : c : \neg p$	$20 \leadsto^{\Diamond} 25$
(26)	$c : \neg p$	$25 \leadsto^{@} 26$

Figure 3: Example 3

The relation $\prec_{\mathcal{B}}$ *in this branch can be represented by the trees in Figure 4, and the single-node tree* 0.

The branch is closed because of nodes 23 and 26. In $\mathcal{B}_{20}$*, node 19 is not blocked by 6, since* $a : \Diamond a : p$ *cannot be mapped to* $c : \Diamond c : p$ *because* c *and* a *are not compatible in* $\mathcal{B}_{20}$ *(*$\Phi_{20}(c) = \emptyset \neq \{p\} = \Phi_{20}(a)$*); therefore, node 19 can be expanded. In the same branch segment, on the contrary, node 20 is blocked by 12, because* $\Phi_{20}(c) = \emptyset = \Phi_{20}(b)$*.*

When the construction proceeds, expanding the non-blocked node 19, and nodes 21–23 are added to the branch, c *and* b *are no more compatible (*$\Phi_{23}(c) = \{p\}$ *while* $\Phi_{23}(b)$ *is still empty), so node 20 is unblocked and it is expanded, producing 24–26 and the branch closes.*

Note moreover that, after the addition of node 23, a *and* c *become compatible, so that in* $\mathcal{B}_{23}$ *node 19 is blocked by 6, and 21–23 are phantom nodes. Since 20 is not a descendant of 19* w.r.t. $\prec_{\mathcal{B}}$*, it is not a phantom, thus it can be expanded.*

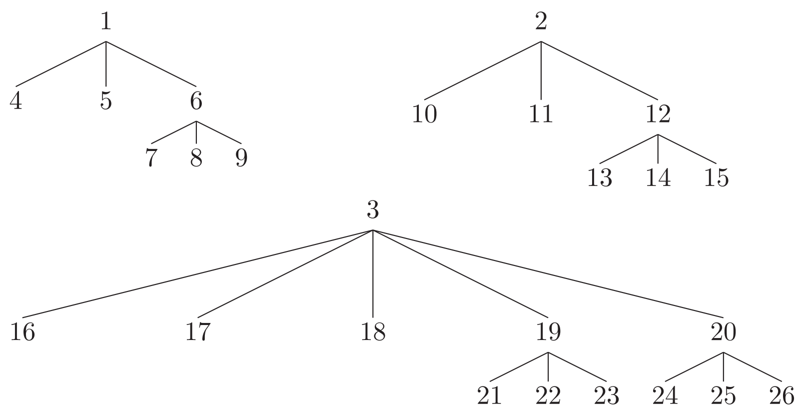

Figure 4: The relation $\prec_{\mathcal{B}}$ on the nodes of the branch of Figure 3

Example 4. *This example shows the need of indirect blocking (restriction* **R3***) to ensure termination. Let*

$$F = a : ((\Box \downarrow x.\Diamond \downarrow y.(x : p \wedge a : \Diamond y)) \wedge \Diamond q)$$

Figure 5 shows a complete branch in a tableau for F.

(1)	$a_0 : F$	
(2)	$a : ((\Box \downarrow x.\Diamond \downarrow y.$ $(x : p \wedge a : \Diamond y)) \wedge \Diamond q)$	$1 \leadsto^{@} 2$
(3)	$a : \Box \downarrow x.\Diamond \downarrow y.$ $(x : p \wedge a : \Diamond y)$	$2 \leadsto^{\wedge} 3$
(4)	$a : \Diamond q$	$2 \leadsto^{\wedge} 4$
(5)	$a : \Diamond b$	$4 \leadsto^{\Diamond} 5$
(6)	$b : q$	$4 \leadsto^{\Diamond} 6$
(7)	$b :\downarrow x.\Diamond \downarrow y.$ $(x : p \wedge a : \Diamond y)$	$(3,5) \leadsto^{\Box} 7$
(8)	$b : \Diamond \downarrow y.(b : p \wedge a : \Diamond y)$	$7 \leadsto^{\downarrow} 8$
(9)	$b : \Diamond b_1$	$8 \leadsto^{\Diamond} 9$
(10)	$b_1 :\downarrow y.(b : p \wedge a : \Diamond y)$	$8 \leadsto^{\Diamond} 10$
(11)	$b_1 : (b : p \wedge a : \Diamond b_1)$	$10 \leadsto^{\downarrow} 11$
(12)	$b_1 : b : p$	$11 \leadsto^{\wedge} 12$
(13)	$b_1 : a : \Diamond b_1$	$11 \leadsto^{\wedge} 13$
(14)	$b : p$	$12 \leadsto^{@} 14$
(15)	$a : \Diamond b_1$	$13 \leadsto^{@} 15$
(16)	$b_1 :\downarrow x.\Diamond \downarrow y.$ $(x : p \wedge a : \Diamond y)$	$(3,15) \leadsto^{\Box} 16$
(17)	$b_1 : \Diamond \downarrow y.(b_1 : p \wedge a : \Diamond y)$	$16 \leadsto^{\downarrow} 17$
(18)	$b_1 : \Diamond b_2$	$17 \leadsto^{\Diamond} 18$
(19)	$b_2 :\downarrow y.(b_1 : p \wedge a : \Diamond y)$	$17 \leadsto^{\Diamond} 19$
(20)	$b_2 : (b_1 : p \wedge a : \Diamond b_2)$	$19 \leadsto^{\downarrow} 20$
(21)	$b_2 : b_1 : p$	$20 \leadsto^{\wedge} 21$
(22)	$b_2 : a : \Diamond b_2$	$20 \leadsto^{\wedge} 22$
(23)	$b_1 : p$	$21 \leadsto^{@} 23$
(24)	$a : \Diamond b_2$	$22 \leadsto^{@} 24$
(25)	$b_2 :\downarrow x.\Diamond \downarrow y.$ $(x : p \wedge a : \Diamond y)$	$(3,24) \leadsto^{\Box} 25$
(26)	$b_2 : \Diamond \downarrow y.(b_2 : p \wedge a : \Diamond y)$	$25 \leadsto^{\downarrow} 26$
(27)	$b_2 : \Diamond b_3$	$26 \leadsto^{\Diamond} 27$
(28)	$b_3 :\downarrow y.(b_2 : p \wedge a : \Diamond y)$	$26 \leadsto^{\Diamond} 28$
(29)	$b_3 : (b_2 : p \wedge a : \Diamond b_3)$	$28 \leadsto^{\downarrow} 29$
(30)	$b_3 : b_2 : p$	$29 \leadsto^{\wedge} 30$
(31)	$b_3 : a : \Diamond b_3$	$29 \leadsto^{\wedge} 31$
(32)	$b_2 : p$	$30 \leadsto^{@} 32$
(33)	$a : \Diamond b_3$	$31 \leadsto^{@} 33$

Figure 5: Example 4.

The relation $\prec_{\mathcal{B}}$ in this branch can be described as follows: the root nodes are 1–4, $4 \prec_{\mathcal{B}} \{5, ..., 8\}$, $8 \prec_{\mathcal{B}} \{9, ..., 17\}$, $17 \prec_{\mathcal{B}} \{18, ..., 26\}$ and $26 \prec_{\mathcal{B}} \{27, ..., 33\}$.[3]

In $\mathcal{B}_{17}$, node 17 is not blocked by 8 because $\Phi_{17}(b) = \{q, p\} \neq \emptyset = \Phi_{17}(b_1)$. And it is not blocked by 8 in $\mathcal{B}_n$ for any $n \geq 23$ either, where $\Phi_n(b) = \{q, p\} \neq \{p\} = \Phi_n(b_1)$.

[3] $n \prec_{\mathcal{B}} \{m_1, ..., m_k\}$ abbreviates $n \prec_{\mathcal{B}} m_1$ and ... $n \prec_{\mathcal{B}} m_k$.

Moreover in $\mathcal{B}_{26}$, node 26 is blocked neither by 8 nor by 17, because $\Phi_{26}(b) = \{q, p\}$, $\Phi_{26}(b_1) = \{p\}$, and $\Phi_{26}(b_2) = \emptyset$.

But in $\mathcal{B}_{33}$, node 26 is blocked by 17, because $\Phi_{33}(b_1) = \{p\} = \Phi_{33}(b_2)$. Therefore, its children w.r.t. $\prec_{\mathcal{B}_{33}}$, i.e. *27–33 are all phantom nodes, and, in particular, node 33 cannot participate, with node 3, to an expansion via the $\Box$ rule.*

Without restriction **R3**, *the construction of the branch would go on forever. In fact, the following nodes could be added:*

(34)	$b_3 :\downarrow x.\Diamond \downarrow y.(x : p \wedge a : \Diamond y)$	$(3, 33) \rightsquigarrow^{\Box} 34$
(35)	$b_3 : \Diamond \downarrow y.(b_3 : p \wedge a : \Diamond y)$	$34 \rightsquigarrow^{\downarrow} 35$

In $\mathcal{B}_{35}$, node 35 would not be blocked, because $\Phi_{35}(b_3) = \emptyset$, while $\Phi_{35}(b_1) = \Phi_{35}(b_2) = \{p\}$. So a sequence of new nodes could be added, with labels obtained from the labels of 27–34, by renaming b_2 with b_3 and b_3 with a new nominal b_4. A neverending story ...

4 Termination

In this section we prove that the tableau calculus defined in Section 3 terminates, *provided that the initial formula is in the $\mathsf{HL}(@, \downarrow) \setminus \downarrow \Box$ fragment.* In this and the following section we always assume that the initial formula is a ground formula in such a fragment.

For the purposes of proving termination and completeness, the main property of the considered fragment is that, if $\Box G$ is a subformula of the initial formula, then it contains no free variables, because it is not in the scope of a binder. As a consequence, for any node label of the form $a : \Box G$, the only nominals occurring in G are top nominals. The first result proved below establishes this fact, along with the standard subformula property.

Definition 7. *Let $\mathcal{B}$ be a tableau branch and $(n_0)\, a_0 : F_0$ its top formula. Then $\mathit{Subf}(\mathcal{B})$ is the set of the subformulae of F_0 and*

$$\mathit{Cmp}(\mathcal{B}) = (\mathit{Subf}(\mathcal{B}) \cap \mathsf{PROP}) \cup \{\Box G \mid \Box G \in \mathit{Subf}(\mathcal{B})\}$$

Lemma 2 (Subformula properties)**.** *For any formula $a : F$ occurring in a branch $\mathcal{B}$ which is not a relational formula, F is an instance of a formula in $\mathit{Subf}(\mathcal{B})$.*

Moreover, if F has the form $\Box G$, then $F \in \mathit{Subf}(\mathcal{B})$. Therefore, for any nominal a, $\Phi_{\mathcal{B}}(a) = \{F \mid a : F \in \mathcal{B}\} \cap \mathit{Cmp}(\mathcal{B})$.

Proof. The proof is an induction on the construction of $\mathcal{B}$, which simultaneously proves the following strongest versions of the two properties: if $(n)\, a : F$ is a node in $\mathcal{B}$ such that $a : F$ is not a relational formula, then for any subformula F' of F:

(A) F' is an instance of a formula in $\mathit{Subf}(\mathcal{B})$, and

(B) if F' has the form $\Box G$, then $F' \in \mathit{Subf}(\mathcal{B})$.

The one-node branch constituting the initial tableau trivially enjoys the required properties. Below, we show that they are preserved by the expansion rules, assuming that $\mathcal{B}$ is obtained from $\mathcal{B}'$ by application of the rule $\mathcal{R}$. We consider different cases according to the rule $\mathcal{R}$, restricting our attention to the node labels which are either added or modified by $\mathcal{R}$.

1. $\mathcal{R}$ is one of the rules $\wedge, \vee, @, \downarrow$, and the node $(n)\, a : F$ is added as an expansion of $(m)\, b : H$.

   (a) If $\mathcal{R}$ is one of the rules $\wedge, \vee, @$, then F is a subformula of H, for which A and B hold by the inductive hypothesis; therefore A and B hold for F too, since any subformula of F is also a subformula of H.

   (b) If $\mathcal{R} = \downarrow$, then H has the form $\downarrow x.H_1$ (for some variable x), and $F = H_1[a/x]$. By the inductive hypothesis, $\downarrow x.H_1$ is an instance of some formula $\downarrow x.H_1' \in \mathit{Subf}(\mathcal{B})$. Since $H_1[a/x]$ is an instance of H_1', which belongs to $\mathit{Subf}(\mathcal{B})$, A holds for F. Moreover, in the fragment $\mathsf{HL}(@, \downarrow) \setminus \downarrow \Box$, no subformula of H_1 has the form $\Box G$, so that B is vacuously true.

2. If $\mathcal{R} = \Diamond$, applied to expand a node $(m)\, b : \Diamond F$, it generates nodes $(k)\, b : \Diamond a$ and $(n)\, a : F$. The label of the first one is a relational formula, so only the second node has to be considered. The same reasoning as in case 1a shows that A and B hold for F.

3. If $\mathcal{R} = \Box$, applied to $(m)\, b : \Box F$ and $(k)\, b : \Diamond a$, it generates the node $(n)\, a : F$. By the inductive hypothesis, $\Box F \in \mathit{Subf}(\mathcal{B})$, hence also all the subformulae of $\Box F$ (including F and its subformulae) are in $\mathit{Subf}(\mathcal{B})$. So, A and B hold for F.

4. Finally, let us consider the case where the equality rule is applied to $\mathcal{B}'$, replacing the nominal c with b, and let $a_0 : F_0$ be the top formula of $\mathcal{B}'$. The top formula of $\mathcal{B}$ is therefore $a_0[b/c] : F_0[b/c]$.

   Let us consider the label $a : F$ of any node n in $\mathcal{B}'$, which is not a relational formula. The label of n in $\mathcal{B}$ is $a[b/c] : F[b/c]$. Any subformula of $F[b/c]$ is obtained from a subformula F' of F by replacing c with b, *i.e.* it has the form $F'[b/c]$. By the inductive hypothesis F' is an instance of a subformula G of F_0, consequently $F'[b/c]$ is an instance of $G[b/c]$, which is a subformula of $F_0[b/c]$.

   Moreover, by the inductive hypothesis, if F' has the form $\Box G$ then it is a subformula of F_0. Therefore, $F'[b/c] = \Box G[b/c]$ is a subformula of $F_0[b/c]$. □

Note that if $\mathcal{B}$ is any branch in a tableau for $F_0(a_1, ...a_n)$, *i.e.* a tableau initialized with $(n_0)\, a_0 : F_0(a_1, ...a_n)$, then the label of n_0 in $\mathcal{B}$ has the form $a_0' : F_0(a_1', ...a_n')$, where a_i' is either a_i or a nominal replacing it. Lemma 2 establishes that any formula $\Box H$ occurring in $\mathcal{B}$ is a subformula of $a_0' : F_0(a_1', ...a_n')$ (and its nominals are top nominals in $\mathcal{B}$, *i.e.* they occur in $a_0' : F_0(a_1', ...a_n')$). As a consequence, the number of universal formulae occurring in $\mathcal{B}$ is bounded by the number of subformulae of $F_0(a_1, ...a_n)$.

In order to prove termination, we first show that, in the forest of trees induced by $\prec_{\mathcal{B}}$ on the nodes of a branch $\mathcal{B}$, any node has a bounded number of siblings. Let us observe that it would not suffice to show that the number of formulae that can label the siblings of a given node is bounded, because, in principle, a given formula might be the label of an infinite number of nodes (a branch is not a set of formulae). In fact, notwithstanding restriction **R1**, distinct node labels can become equal by effect of substitution.

The relation $\rhd$, which is defined next, introduces an order on the siblings of a given node.

Definition 8. *Let n, m and k be nodes in $\mathcal{B}$.*

- *If* $n \rightsquigarrow^{\mathcal{R}} m$*, for* $\mathcal{R} = \wedge, \vee, @, \downarrow$*, then* $n \rhd m$*;*
- *If* $(m, n) \rightsquigarrow^{\Box} k$*, and* n *is the minor premiss of the inference, then* $n \rhd k$

$\rhd^*$ *is the reflexive and transitive closure of* $\rhd$*. If* $n \rhd^* m$*, we say that* n produces m*.*

It is worth noticing that the property that n and m are siblings *w.r.t.* $\prec_{\mathcal{B}}$ is implied by $n \rhd m$ (see Definition 4), but obviously the two relations are not equivalent. In fact, $\rhd$ is not a symmetric relation.

In what follows, $|F|$ is the size of the formula F, counted as the number of symbols in F. Below, we tacitly exploit the trivial fact that the size of the top formula of any branch is the same as the size of the initial formula of the tableau.

Lemma 3. *Let* n *be a node in a branch* $\mathcal{B}$ *of a tableau for a formula* F*. Then the cardinality of* $\Sigma(n) = \{m \mid n \rhd^* m\}$ *is bounded by a function of* $|F|$*.*

Proof. It is worth pointing out again that, in principle, $\Sigma(n)$ may contain nodes labelled by a same formula, so the reasoning is not as simple as it would be if dealing with sets of formulae. However, as shown below, the label of any sibling of n has a *matrix* taken from a bounded stock of formulae, that is built in the language of the branch *at the time* n *is added to it.* Node labels with the same matrix are always equal, at any construction stage of the branch, so that the cardinality of $\Sigma(n)$ is bounded by the number of such possible *matrices.*

Any branch $\mathcal{B}$ in a tableau is the last element of a sequence of branches, where the first one is the initial tableau, and each of the others is obtained from the previous one by application of an expansion rule. Such a sequence will be called the *sequence of branches leading to* $\mathcal{B}$.

Let n be any fixed node in a tableau branch $\mathcal{B}$. We shall use the following notations:

1. $\mathcal{B}_1$ is the first branch where n occurs, in the sequence of branches leading to $\mathcal{B}$, and the sequence $\mathcal{B}_1, \mathcal{B}_2, ..., \mathcal{B}_p = \mathcal{B}$ denotes the sequence of branches which, starting from $\mathcal{B}_1$, leads to $\mathcal{B}$; *i.e.* it is the subsequence of the sequences of branches leading to $\mathcal{B}$ which starts from $\mathcal{B}_1$.

2. $label_{\mathcal{B}_i}(k)$ is the label of the node k in the branch $\mathcal{B}_i$. In the proof we need in fact to refer to node labels in different branches.

3. For $1 \leq i \leq p$, σ_i is the composition of the sequence of substitutions applied in the sequence $\mathcal{B}_1, ..., \mathcal{B}_p$, by means of the equality rule, up to $\mathcal{B}_i$ included. Consequently, for each $i > 0$, $label_{\mathcal{B}_i}(n) = \sigma_i(label_{\mathcal{B}_1}(n))$.

4. M_n is the set containing all the nominals occurring in $label_{\mathcal{B}_1}(n)$ and all the top nominals in $\mathcal{B}_1$.

5. S_n is the set containing all the subformulae of some universal subformula of the top formula of $\mathcal{B}_1$, and all the ground formulae that can be obtained from a subformula of $label_{\mathcal{B}_1}(n)$ by replacing free variables with elements of M_n.

6. $\mathcal{F}_n = \{a : F \mid a \in M_n \text{ and } F \in S_n\} \cup \{a : \Diamond b \mid a, b \in M_n\}$. Any element of $\mathcal{F}_n$ will be called a *matrix.* Note that only nominals in M_n may occur in a matrix.

7. N is the size of the top formula of $\mathcal{B}$, which is obviously equal to the size of the initial formula of the tableau.

It is easy to see that $|M_n| \leq N$. In fact, if $a_0 : F_0$ is the top formula of $\mathcal{B}_1$, $|M_n|$ cannot exceed 1 (for the outermost nominal in $label_{\mathcal{B}_1}(n)$) + the sum of the number of top nominals and the number of variables occurring in $a_0 : F_0$ (by Lemma 2). Such a sum is not greater than $N - 1$, *i.e.* the number of symbols in F_0 plus 1 for the outermost nominal a_0.

Moreover, $|S_n| \leq N + N^3$. In fact, the number of subformulae of some universal subformula of the top node of $\mathcal{B}_1$ is not greater than N. The number of subformulae of $label_{\mathcal{B}_1}(n)$ is not greater than N either (by Lemma 2); each of them contains at most N free variables, and each free variable can be instantiated in at most N different ways (the number of elements of M_n). Therefore there are no more than N^3 ground formulae that can be obtained from a subformula of $label_{\mathcal{B}_1}(n)$ by replacing state variables with elements of M_n.

Consequently, $|\mathcal{F}_n| \leq 2N^2 + N^4$: for each formula $a : F$ with $a \in M_n$ and $F \in S_n$ there are no more than N choices for the nominal a and no more than $N + N^3$ choices for the formula F; therefore the cardinality of $\{a : F \mid a \in M_n \text{ and } F \in S_n\}$ is bounded by $N^2 + N^4$. And formulae of the form $a : \Diamond b$ with $a, b \in M_n$ can be built in at most N^2 different ways.

Let m be any node in $\Sigma(n)$, *i.e.* $n \rhd^* m$. We say that an element F of $\mathcal{F}_n$ is the *matrix* of m in $\mathcal{B}_i$ if $label_{\mathcal{B}_i}(m) = \sigma_i(F)$. F is the matrix of m if it is the matrix of m in all $\mathcal{B}_i$ where m occurs ($i = 1, ..., p$). If two nodes m_1 and m_2 have the same matrix, then obviously for all $i = 1, ..., p$, $label_{\mathcal{B}_i}(m_1) = label_{\mathcal{B}_i}(m_2)$.

We first prove that:

(A) the label of any node in $\Sigma(n)$ has a matrix in $\mathcal{F}_n$. *I.e.* if $m \in \Sigma(n)$, then for all $i = 1, ..., p$, $label_{\mathcal{B}_i}(m) = \sigma_i(F)$ for some $F \in \mathcal{F}_n$.

The proof is by induction on i. If $i = 1$ then necessarily $m = n$, $\sigma_1 = \emptyset$ and $label_{\mathcal{B}_1}(n) \in \mathcal{F}_n$. Otherwise, if $i > 1$, we consider different cases according to the rule applied to obtain $\mathcal{B}_i$ from $\mathcal{B}_{i-1}$. Note that in all cases, except for the first one, $\sigma_i = \sigma_{i-1}$ and $label_{\mathcal{B}_i}(m) = label_{\mathcal{B}_{i-1}}(m)$ for any node m occurring in $\mathcal{B}_{i-1}$. Therefore, in all but the first case, the thesis must be proved only for the newly added nodes, and in the treatment of such cases we assume that m is any node in $\mathcal{B}_i$ which does not belong to $\mathcal{B}_{i-1}$.

(=) Let m be any node in $\Sigma(n)$. By the induction hypothesis, there is a formula $F \in \mathcal{F}_n$ such that $label_{\mathcal{B}_{i-1}}(m) = \sigma_{i-1}(F)$. If $\mathcal{B}_{i-1}$ is expanded by means of the equality rule replacing a with b, then $label_{\mathcal{B}_i}(m) = (\sigma_{i-1}(F))[a/b]$. Since $\sigma_i = \sigma_{i-1} \circ \{a/b\}$, $label_{\mathcal{B}_i}(m) = \sigma_i(F)$, therefore F is the matrix of m in $\mathcal{B}_i$.

($\wedge, \vee$) Let $n \rhd^* k \leadsto^{\mathcal{R}} m$, for $\mathcal{R} \in \{\wedge, \vee\}$, with $label_{\mathcal{B}_i}(k) = a : F'_1 \star F'_2$ (for $\star \in \{\wedge, \vee\}$), and $label_{\mathcal{B}_i}(m) = a : F'_j$ ($j = 1, 2$). By the induction hypothesis, since k occurs in $\mathcal{B}_{i-1}$, $a : F'_1 \star F'_2 = \sigma_{i-1}(c : F_1 \star F_2) = \sigma_i(c : F_1 \star F_2)$ for some $c : F_1 \star F_2 \in \mathcal{F}_n$, *i.e.* $c \in M_n$ and $F_1 \star F_2 \in S_n$. Since S_n is closed *w.r.t.* subformulae, $F_j \in S_n$, therefore $c : F_j \in \mathcal{F}_n$. Since $label_{\mathcal{B}_i}(m) = a : F'_j = \sigma_i(c : F_j)$, $c : F_j$ is the matrix of m in $\mathcal{B}_i$.

(@) Let $n \rhd^* k \leadsto^{@} m$, with $label_{\mathcal{B}_{i-1}}(k) = label_{\mathcal{B}_i}(k) = a : b : F'$ and $label_{\mathcal{B}_i}(m) = b : F'$. By the induction hypothesis, $a : b : F' = \sigma_{i-1}(c : d : F) = \sigma_i(c : d : F)$ for some

$c : d : F \in \mathcal{F}_n$. Since $d : F \in S_n$, $d \in M_n$ and $F \in S_n$. Therefore also $d : F \in \mathcal{F}_n$. So, since $label_{\mathcal{B}_i}(m) = b : F' = \sigma_i(d : F)$, $d : F$ is the matrix of m in $\mathcal{B}_i$.

($\downarrow$) Let $n \rhd^* k \leadsto^{\downarrow} m$, with $label_{\mathcal{B}_i}(k) = a :\downarrow x.F'$ and $label_{\mathcal{B}_i}(m) = a : F'[a/x]$. By the induction hypothesis, $a :\downarrow x.F' = \sigma_{i-1}(c :\downarrow x.F) = \sigma_i(c :\downarrow x.F)$ for some $c :\downarrow x.F \in \mathcal{F}_n$. Since $\downarrow x.F \in S_n$, any instance of F replacing x with a nominal in M_n is in S_n. In particular $F[c/x] \in S_n$. Since $c \in M_n$ and $a : F'[a/x] = \sigma_i(c : F[c/x])$, $c : F[c/x]$ is the matrix of m in $\mathcal{B}_i$.

($\Box$) Let $n \rhd^* k$ and $(k', k) \leadsto^{\Box} m$, with $label_{\mathcal{B}_{i-1}}(k') = label_{\mathcal{B}_i}(k') = a : \Box G$, $label_{\mathcal{B}_{i-1}}(k) = label_{\mathcal{B}_i}(k) = a : \Diamond b$ and $label_{\mathcal{B}_i}(m) = b : G$. By Lemma 2, $\Box G \in S_n$. Therefore also $G \in S_n$. By the induction hypothesis, $a : \Diamond b = \sigma_i(c : \Diamond d)$ for some $c : \Diamond d \in \mathcal{F}_n$, *i.e.* $b = \sigma_i(d)$ for $d \in M_n$. Therefore $d : G \in \mathcal{F}_n$ and, since $b : G = \sigma_i(d : G)$, $d : G$ is the matrix of m in $\mathcal{B}_i$.

Next we observe that:

(B) for any pair of nodes m, k and any branch $\mathcal{B}$, if $m \rhd k$ then m is a phantom node in $\mathcal{B}$ if and only if k is a phantom in $\mathcal{B}$. Consequently, also if $m \rhd^* k$ then m is a phantom in $\mathcal{B}$ if and only if k is a phantom in $\mathcal{B}$. And, for any branch $\mathcal{B}$, either all elements of $\Sigma(n)$ are phantom nodes in $\mathcal{B}$ or none of them is a phantom in $\mathcal{B}$.

We can now prove that the cardinality of $\Sigma(n)$ is bounded by $h(N)$, where $h(N) = 2N^2 + N^4$ is the cardinality of $\mathcal{F}_n$. Let us assume, by absurd, that $\Sigma(n)$ has more than $h(N)$ elements. Then, by A, there are at least two distinct elements m_1 and m_2 in $\Sigma(n)$ which have the same matrix F. We may assume *w.l.g.* that $n \leq m_1 < m_2$. Let $\mathcal{B}_k$ be the first branch in the sequence $\mathcal{B}_1, ..., \mathcal{B}_p$ where m_2 occurs. Since $n < m_2$, there is a node $k \in \Sigma(n)$ such that $n \rhd^* k \rhd m_2$. Since k produces a node, its label has not the form $a : \Box G$, so that, if k were a phantom in $\mathcal{B}_{k-1}$, its expansion would violate restriction **R3**. Therefore k is not a phantom in $\mathcal{B}_{k-1}$, and consequently, by B, m_1 is not a phantom in $\mathcal{B}_{k-1}$ either. But $label_{\mathcal{B}_k}(m_2) = \sigma_k(F) = \sigma_{k-1}(F) = label_{\mathcal{B}_{k-1}}(m_1)$ ($\sigma_k = \sigma_{k-1}$ because, clearly, $\mathcal{B}_{k-1}$ has not been expanded by means of the equality rule, which does not add new nodes to the branch). Therefore, the addition of m_2 to $\mathcal{B}_{k-1}$ violates restriction **R1**. $\Box$

The next result states that the forest of trees induced by $\prec_{\mathcal{B}}$ on any branch $\mathcal{B}$ has a bounded number of trees, and each tree has a bounded width.

Lemma 4. *Let $\mathcal{B}$ be a branch in a tableau for F.*

- *The number of root nodes in $\mathcal{B}$ is bounded by a function of $|F|$.*
- *For any node n of $\mathcal{B}$, the cardinality of the set of the children of n,* i.e. *the set $\Gamma_{\mathcal{B}}(n) = \{m \mid n \prec_{\mathcal{B}} m\}$, is bounded by a function of $|F|$.*

Proof. The first item follows immediately from Lemma 3, since all root nodes are produced by the top node. The second item also follows from Lemma 3 by observing that $n \prec_{\mathcal{B}} m$ if and only if either $n \leadsto^{\Diamond} (p, q)$ and m is one of p, q, or else there is a node s such that $n \prec_{\mathcal{B}} s$ and $s \rhd m$. Equivalently, $n \prec_{\mathcal{B}} m$ if and only if for some nodes p, q we have $n \leadsto^{\Diamond} (p, q)$ and either $p \rhd^* m$ or $q \rhd^* m$. $\Box$

Next, we show that any tree in the forest induced by $\prec_{\mathcal{B}}$ on the nodes of $\mathcal{B}$ has a bounded depth. To this aim, we define an equivalence relation among node labels.

Definition 9. *Let F be a* ground *formula containing exactly the nominals $a_1, .., a_n$. Let $W = \{w_1, w_2, ..., w_n\}$ be a set of fresh variables, and let μ be a bijection from $\{a_1, .., a_n\}$ onto W. A skeleton F_S for F is the formula obtained from F by replacing every nominal a_j by $\mu(a_j)$.*

Since a skeleton for a given formula is unique up to variable renaming, we shall speak of "the skeleton" of a formula, and we consider two skeletons identical if they only differ for the choice of variable names.

Example 5. *The formulae $a_1 : \Diamond \downarrow x.(\Box x \wedge \neg a_1)$ and $a_2 : \Diamond \downarrow x.(\Box x \wedge \neg a_2)$ have the same skeleton $w_1 : \Diamond \downarrow x.(\Box x \wedge \neg w_1)$. However, $a_1 : \Diamond \downarrow x.(\Box x \wedge \neg a_2)$ has a different skeleton, that is $w_1 : \Diamond \downarrow x.(\Box x \wedge \neg w_2)$.*

Definition 10. *Let $\mathcal{B}$ be a tableau branch, and F_1, F_2 two node labels in $\mathcal{B}$. Then $F_1 \approx_{\mathcal{B}} F_2$ if and only if:*

1. *F_1 and F_2 have the same skeleton $F_S(w_1, ..., w_n)$;*
2. *if μ_i ($i = 1, 2$) is the bijection from the nominals in F_i onto $\{w_1, ..., w_n\}$ establishing that F_S is the skeleton of F_i, then for all $j = 1, ..., n$, $\mu_1^-(w_j)$ and $\mu_2^-(w_j)$ are compatible in $\mathcal{B}$.*

The relation $\approx_{\mathcal{B}}$ is obviously an equivalence relation on the (ground) formulae occurring as node labels in $\mathcal{B}$.

The next result establishes a bound on the number of possible skeletons for node labels in a tableau branch.

Lemma 5. *Let $\mathcal{B}$ be a branch whose top formula is F_0. The number of distinct possible skeletons for blockable node labels in $\mathcal{B}$ is bounded by $|F_0|^4$.*

Proof. By Lemma 2, for any blockable node label $a : F$ in $\mathcal{B}$, F is an instance of a subformula of F_0 (in fact, relational formulae are not blockable).

Obviously, different instances of the same subformula $F(x_1, ..., x_n, a_1, ..., a_k)$ (where the free variables and the nominals occurring in F are made explicit) can have different skeletons: if $F_S(w_1, ..., w_n, w_{n+1}, ..., w_{n+k})$ is the skeleton of any instance of F, then each w_j, for $1 \leq j \leq n$, can be either equal to some of the other w_i or different from them all, *i.e.* there are $n+k$ choices for each w_j. However, there are no more than $n \cdot (n+k)$ different skeletons for instances of F, and $n \cdot (n + k) \leq N^2$, where $N = |F_0|$. Therefore, there are at most N^3 possible skeletons for the body of a node label (N^2 different skeletons for each of the N subformulae of F_0).

The outermost nominal a can either be a nominal which occurs in F or a different one. Since the number of nominals occurring in F is bounded by $N - 1$ (the number of symbols of F_0 excluding the outermost satisfaction symbol), there are N choices for the outermost nominal. Hence, the number of different skeletons for a node label in $\mathcal{B}$ is bounded by N^4. □

Lemma 6. *Let $\mathcal{B}$ be a branch and N the size of its top formula. The maximal number of equivalence classes* w.r.t. $\approx_{\mathcal{B}}$ *of blockable node labels is bounded by a function $K(N)$.*

Therefore the size of any set S of blockable formulae which may occur in a tableau branch $\mathcal{B}$, and such that for any pair of its elements $F, G \in S$, F is not blocked by G in $\mathcal{B}$, is bounded by $K(N)$.

Proof. Let $F_s(w_1, ..., w_k)$ be the skeleton of a node label in $\mathcal{B}$, and ν the cardinality of $Cmp(\mathcal{B})$ (see Definition 7). By Lemma 2, for any nominal a, $\Phi_{\mathcal{B}}(a) = \{F \mid a : F \in \mathcal{B}\} \cap Cmp(\mathcal{B})$. Therefore, there are at most $2^{k \cdot \nu} \leq 2^{N^2}$ node labels in $\mathcal{B}$ sharing the same skeleton $F_s(w_1, ..., w_k)$ but pairwise not equivalent *w.r.t.* $\approx_{\mathcal{B}}$. By Lemma 5, the number of different skeletons for blockable node labels in $\mathcal{B}$ is N^4. Therefore, the number of equivalences classes *w.r.t.* $\approx_{\mathcal{B}}$ of blockable node labels, and consequently the cardinality of the set S, is bounded by $N^4 \cdot 2^{N^2}$. □

Definition 11. *A chain in a branch $\mathcal{B}$ is a sequence of nodes $n_1, n_2...$ such that for all i: $n_i \prec_{\mathcal{B}} n_{i+1}$. If a chain $n_1, ..., n_k$ is finite and n_1 is a root node, we say that it is the maximal chain leading to n_k.*

We recall that, by Lemma 1.1 for any given node n there is exactly one maximal chain leading to n.

Lemma 7. *Let $\mathcal{B}$ be a tableau branch and N the size of its top formula. Then for any chain*

$$n_1 \prec_{\mathcal{B}} n_2 \prec_{\mathcal{B}} .... \prec_{\mathcal{B}} n_k$$

$k \leq K(N) + 1$, where $K(N)$ is the bound given by Lemma 6.

Proof. We note beforehand that if $n \prec_{\mathcal{B}} m$, then:

- $n < m$ in the branch;
- the label of n is a blockable formula, therefore in any chain $n_1 \prec_{\mathcal{B}} n_2 \prec_{\mathcal{B}} .... \prec_{\mathcal{B}} n_k$, for all $i = 1, ..., k-1$, the label of n_i is a blockable formula (Lemma 1.2).

Let us assume that a branch $\mathcal{B}$ contains a chain $n_1 \prec_{\mathcal{B}} n_2 \prec_{\mathcal{B}} .... \prec_{\mathcal{B}} n_K \prec_{\mathcal{B}} n_{K(N)+1}$. We show that such a chain cannot be extended.

If the label of $n_{K(N)+1}$ is not a blockable formula, the chain cannot be extended, by the definition of $\prec_{\mathcal{B}}$.

Otherwise, for all $i = 1, ..., K(N) + 1$, the label of n_i is a blockable formula, so, by Lemma 6, there are at least two indexes $1 \leq i < j \leq K(N) + 1$ such that n_i blocks n_j in $\mathcal{B}$. If $j = K(N) + 1$, then $n_{K(N)+1}$ is directly blocked and cannot be expanded, by restriction **R4**. Otherwise, if $j \leq K(N)$, then n_j is directly blocked by n_i; so, $n_{K(N)+1}$ is a phantom in $\mathcal{B}$ and it cannot be expanded either, by restriction **R3**. Hence, in any case, no node $n_{K(N)+2}$ such that $n_{K(N)+1} \prec n_{K(N)+2}$ can be generated. □

It is worth noticing that the above proof would be exactly the same if ancestor blocking (relying on the relation $\prec_{\mathcal{B}}$) were adopted, since it reasons on chains of nodes. However restriction **R4** is actually stricter than ancestor blocking: *any* node n may block any other node m, provided that $n < m$

Theorem 1 (Termination)**.** *Every tableau branch has a bounded depth and tableau construction always terminates.*

Proof. By Lemmas 1.1, 4 and 7, the nodes of a branch $\mathcal{B}$ are arranged by $\prec_{\mathcal{B}}$ in a bounded sized set of trees, each of which has bounded width and bounded depth. Hence any tableau branch $\mathcal{B}$ has a number of nodes that is bounded by the size of the initial formula.

Since every rule (except for the equality rule) adds some node to the current branch, the only reason why tableau construction might not terminate is that the equality rule is applied an infinite number of times. But this is absurd, since every application of such a rule reduces the number of nominals occurring in the branch. □

5 Completeness

Completeness will be proved in the standard way, by showing how to define a model of the initial formula from a complete and open tableau branch. However, for the calculus introduced in this work, the fact that the labels of blocked and blocking nodes are not identical must be taken into account. A model cannot be simply built from a set of worlds consisting of equivalence classes of nominals, where two nominals are in the same class whenever some blocking mapping maps one to the other. Consider in fact, for instance, example 1: although the branch is open, a model cannot directly be extracted from it, making b and c denote the same world: although c and b are in some sense identified by the mapping used to block node 17, they cannot denote the same world in the model, because of the presence of node 13, that forces them to denote distinct worlds.

Thus, we follow a different approach, showing that a (possibly infinite) model can be built out of a complete and open branch $\mathcal{B}$ by means of a preliminary infinitary extension $\mathcal{N}^{\infty}_{\mathcal{B}}$ of a subset of $\mathcal{B}$.[4]

Let $\mathcal{C}$ be the set containing all the nodes of $\mathcal{B}$ whose label has the form $a : F$, for $F \in Cmp(\mathcal{B})$ (see Definition 7), and $\mathcal{N}_0$ be the union of $\mathcal{C}$ and the set of the non-phantom nodes of $\mathcal{B}$. We inductively construct an infinite sequence of finite extensions of $\mathcal{N}_0$: $\mathcal{N}_0 \subseteq \mathcal{N}_1 \subseteq \mathcal{N}_2, \ldots$. Each $\mathcal{N}_i$ is associated an order $<_i$ and a set of triples $\mathbf{B}_i$, that will be called the *blocking relation* for $\mathcal{N}_i$. At each stage in the construction, new nodes can be added to obtain $\mathcal{N}_{i+1}$ from $\mathcal{N}_i$. Each of them "corresponds" to some node $n \in \mathcal{N}_0$ (its label is a renaming of $label(n)$). If a node added at stage i corresponds to the node $n \in \mathcal{N}_0$, it will be denoted by n^i. For the sake of generality, a node $n \in \mathcal{N}_0$ is identified with n^0. Moreover, each stage $i > 0$ introduces at most a new nominal, for which the meta-notation b^i will be used. For any i, each triple in the blocking relation $\mathbf{B}_i$ for $\mathcal{N}_i$ has the form (n^q, m, π), where n^q and m are distinct nodes in $\mathcal{N}_i$ and π is an injective function from non-top nominals occurring in m to non-top nominals occurring in n^q. We say that a nominal b is a *witness* in $\mathcal{N}_i$ of a node $(n)\, a : \Diamond F \in \mathcal{N}_i$ if $\mathcal{N}_i$ contains nodes labelled, respectively, by $a : \Diamond b$ and $b : F$.

The next subsection details the construction of the sets $\mathcal{N}_i$ and the corresponding relations $<_i$ and $\mathbf{B}_i$, while Subsection 5.2 establishes some important properties of such sets and uses them in order to build the required model.

[4]It is worth noticing that if a formula of the considered hybrid fragment has a model, then it has also a finite model, since this holds for guarded logic [7].

5.1 Unravelling the blockings

Let $\mathcal{B}$ be a complete and open tableau branch. Below, we show how to construct the sequence of sets of nodes $\mathcal{S}_\mathcal{B} : \mathcal{N}_0 \subseteq \mathcal{N}_1 \subseteq \mathcal{N}_2 ...$ and the associated order $<_i$ and relation $\mathbf{B}_i$, such that $\mathcal{N}_0$ is the union of the set $\mathcal{C}$ previously defined and the set of non-phantom nodes in $\mathcal{B}$ (and thus includes the top node of $\mathcal{B}$). The construction also shows that the following *invariants* are satisfied, for any i:

1. if $(n^q, m, \pi) \in \mathbf{B}_i$ $(q \geq 0)$, then:
   (a) $m \in \mathcal{N}_0$ and m is not blocked in $\mathcal{B}$;
   (b) π is an injective mapping from non-top nominals to non-top nominals modifying only nominals occurring in the label of m;
   (c) the formulae labelling n^q and m are blockable formulae and $\pi(label(m)) = label(n^q)$;
2. for any node $n^q \in \mathcal{N}_i$, if n^q has no witness in $\mathcal{N}_i$, then $(n^q, m, \pi) \in \mathbf{B}_i$, for some m and π.

The elements $\mathcal{N}_i$ of $\mathcal{S}_\mathcal{B}$, and the associated relations $<_i$ and $\mathbf{B}_i$, are defined inductively as follows. The proof that the above stated invariants hold goes along with the inductive construction of the sets.

Base: $i = 0$**.** $\mathcal{N}_0$ is the union of the set of non-phantom nodes in $\mathcal{B}$ and the set $\mathcal{C} = \{n \in \mathcal{B} \mid label(n) = a : F \text{ and } F \in Cmp(\mathcal{B})\}$, $<_0 = <$ (where $<$ is the total order on nodes in the sequence $\mathcal{B}$), and

$$\mathbf{B}_0 = \{(n, m, \pi) \mid n \text{ is blocked by } m \text{ in } \mathcal{B} \text{ via the mapping } \pi\}$$

Obviously, all the invariants hold here.

Inductive Step $(i > 0)$**.** We assume that the invariants hold for $i - 1$.

(Case 1.) If $\mathbf{B}_{i-1} = \emptyset$ then $\mathcal{N}_i = \mathcal{N}_{i-1}$, $\mathbf{B}_i = \emptyset = \mathbf{B}_{i-1}$ and $<_i = <_{i-1}$.

(Case 2 2.) Otherwise, let n^p be the first node in $\mathcal{N}_{i-1}$, according to the order $<_{i-1}$, such that $(n^p, m, \pi) \in \mathbf{B}_{i-1}$ for some $m \in \mathcal{N}_0$ and mapping π. Let

$$label(n^p) = a_0 : \Diamond F(a_1, ..., a_k)$$

where $a_1, ..., a_k$ are all the non-top nominals occurring in F. By the invariant 1a, $m \in \mathcal{N}_0$ and it is not blocked in $\mathcal{B}$, and by the invariants 1b and 1c, $label(m)$ has the form $c_0 : \Diamond F(c_1, ..., c_k)$ where for $j = 0, ..., k$, $\pi(c_j) = a_j$. *I.e.* , π is a subset of $\{a_0/c_0, ..., a_k/c_k\}$. By an abuse of notation, we shall however denote π by $\{a_0/c_0, ..., a_k/c_k\}$ itself, although, possibly, for some a_i, $a_i = c_i$. Then there are nodes k_1 and k_2 in $\mathcal{B}$, such that $m \leadsto^\Diamond (k_1, k_2)$, since m is not blocked in $\mathcal{B}$ and $\mathcal{B}$ is complete. Such nodes k_1 and k_2 are not phantom nodes in $\mathcal{B}$, thus they belong to $\mathcal{N}_0$. As a consequence, they belong to $\mathcal{N}_{i-1} \supseteq \mathcal{N}_0$.

Let us assume that $label(k_1) = c_0 : \Diamond b$ and $label(k_2) = b : F(c_1, ..., c_k)$, and let b^i be a fresh nominal. A mapping θ_i, that will guide the construction of the new nodes of $\mathcal{N}_i$, is then defined as follows:

- $\theta_i(c_j) = \pi(c_j)$;
- If $b \notin \{c_0, ..., c_k\}$, then $\theta_i(b) = b^i$.
- $\theta_i(d) = d$ if $d \notin \{b, c_0, ..., c_k\}$

In other words:

$$\theta_i = \begin{cases} \{a_0/c_0, ..., a_k/c_k\} & \text{if } b \in \{c_0, ..., c_k\} \\ \{a_0/c_0, ..., a_k/c_k, b^i/b\} & \text{otherwise} \end{cases}$$

It is worth noticing that b is a witness of m. At the time it was created, by an application of the $\Diamond$ rule, it was obviously fresh *w.r.t.* to the current branch, but it may subsequently have been replaced by some c_i by the equality rule. We observe moreover that θ_i is injective.

$\mathcal{N}_i$ is defined as the union of $\mathcal{N}_{i-1}$ with

$$\{(k^i)\,\theta_i(G) \mid (k)\,G \in \mathcal{N}_0 \text{ and no node in } \mathcal{N}_{i-1} \text{ labels } \theta_i(G)\}$$

$\mathcal{N}_i$ is thus obtained from $\mathcal{N}_{i-1}$ by addition of a finite number of nodes k^i, where each k^i *corresponds* to a node k in $\mathcal{N}_0$ (its label is a "copy" of $label(k)$ modulo the renaming θ_i).

In particular, $\mathcal{N}_i$ contains, for some nominal d and some $j \geq 0$ (possibly, $j = i$): $(k_1^j)a_0 : \Diamond d$ and $(k_2^j)d : F(a_1, ..., a_k)$ (where either $d = b^i$ is a fresh nominal or $d = a_j$ for some j). Hence, $\mathcal{N}_i$ has a witness for $(n^p)a_0 : \Diamond F(a_1, ..., a_k)$.

The order $<_i$ on nodes in $\mathcal{N}_i$ is the extension of $<_{i-1}$ where $n^j <_i m^i$ if $j < i$, and $n^i <_i m^i$ if $n <_0 m$.

Next, the set $\mathbf{B}_i$ is defined, representing the blocking relation in $\mathcal{N}_i$. In the sequel, if F is a formula and γ is a mapping from nominals to nominals, the restriction of γ to the nominals occurring in F, $\gamma\mid_F$, is:

$$\gamma\mid_F(a) = \begin{cases} \gamma(a) & \text{if } a \text{ occurs in } F \\ a & \text{otherwise} \end{cases}$$

The notation $G(d_1, ..., d_n)$ will sometimes be used to denote a formula G where some of the non-top nominals $d_1, ..., d_n$ may occur (beyond other nominals). If π is a mapping that is the identity for nominals not in $\{d_1, ..., d_n\}$, the application of π to G will be denoted by $G(\pi(d_1), ..., \pi(d_n))$.

Now, let S be the set of all the new nodes

$$(q^i)\,\theta_i(G(c_0, c_1, ..., c_k, b)) = (q^i)\,G(a_0, a_1, ..., a_k, \theta_i(b))$$

added at stage i and such that G is a blockable formula with no witness in $\mathcal{N}_i$.

For any node $q^i \in S$, a *blocking node* β_{q^i} and *blocking mapping* μ_{q^i} are defined. Let us consider any $q^i \in S$, with label $\theta_i(G(c_0, c_1, ..., c_k, b))$. Since q^i is added at stage i, there is a node $m \in \mathcal{N}_0$ labelled by $G(c_0, c_1, ..., c_k, b)$, such that q^i corresponds to m. Such a node m has no witness in $\mathcal{N}_0$. In fact, let us assume that $label(m) = G(c_0, c_1, ..., c_k, b)$ has the form $d : \Diamond F$. If m had a witness d' in $\mathcal{N}_0$, then $\mathcal{N}_0$ would contain nodes labelled by $d : \Diamond d'$ and $d' : F$. As a consequence, $\mathcal{N}_i$ would contain nodes labelled by $\theta_i(d) : \Diamond\theta_i(d')$ and $\theta_i(d') : \theta_i(F)$. Since $label(q^i) = \theta_i(G(c_0, c_1, ..., c_k, b)) = \theta_i(d) : \Diamond\theta_i(F)$, then $\theta_i(d')$

would be a witness for q^i in $\mathcal{N}_i$, contradicting the initial hypothesis that q^i has no witness in $\mathcal{N}_i$.

Since m has no witness in $\mathcal{N}_0$, m is blocked in $\mathcal{B}$, thus $\mathbf{B}_0$ contains a triple of the form (m, m', σ). Consequently, $\sigma(label(m')) = label(m)$ and $\theta_i(label(m)) = label(q^i)$. Then we set:

- $\beta_{q^i} = m'$;
- $\mu_{q^i} = (\theta_i \circ \sigma) \mid_{label(m')}$ (see Figure 6).

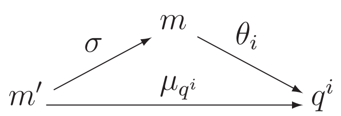

Figure 6: The construction of the mapping μ_{q^i} matching $m' = \beta_{q^i}$ to q^i.

We observe that, since the invariants 1a, 1b and 1c hold for $\mathcal{N}_{i-1}$:

1a) $m' = \beta_{q^i}$ is not blocked in $\mathcal{B}$;

1b) Since both θ_i and σ are injective, μ_{q^i} is injective too. Moreover, μ_{q^i} modifies only nominals occurring in $label(\beta_{q^i})$ by construction;

1c) $label(q^i)$ is a blockable formula, and $(\mu_{q^i})(label(\beta_{q^i})) = (\mu_{q^i})(label(m')) = (\theta_i \circ \sigma)(label(m')) = \theta_i(\sigma(label(m'))) = \theta_i(label(m)) = label(q^i)$.

We then define:

$$\mathbf{B}_i = (\mathbf{B}_{i-1} \ \setminus \ \{(n^p, m, \pi)\}) \cup \{(q^i, \beta_{q^i}, \mu_{q^i} \mid q^i \in S\}$$

In other words, $\mathbf{B}_i$ is obtained from $\mathbf{B}_{i-1}$ by eliminating the triple (n^p, m, π) (since n^p has a witness in $\mathcal{N}_i$), and adding all the triples $(q^i, \beta_{q^i}, \mu_{q^i})$ for any new node without witness in $\mathcal{N}_i$

The three invariants 1a, 1b and 1c still hold for $\mathcal{N}_i$, by the previous observation, and the invariant 2 holds by construction.

Finally, if $\mathcal{B}$ is a complete and open branch, the possibly infinite set of nodes $\mathcal{N}^\infty_\mathcal{B}$ is defined by:

$$\mathcal{N}^\infty_\mathcal{B} = \bigcup_{i \in \mathbb{N}} \mathcal{N}_i$$

We observe that

$$\bigcap_{i \in \mathbb{N}} \mathbf{B}_i = \emptyset$$

and therefore, by the invariant 2, every node labelled by a blockable formula has a witness in $\mathcal{N}^\infty_\mathcal{B}$.

5.2 Model construction

The proof that if $\mathcal{B}$ is a complete and open tableau branch for F then F is satisfiable, exploits the construction of a model of $\mathcal{N}_{\mathcal{B}}^{\infty}$. In order to build such a model, some properties of the sets $\mathcal{N}_i$ and their associated relations have to be preliminarily proved.

In what follows, we shall sometimes write $a : F \in \mathcal{N}_i$ to mean that $\mathcal{N}_i$ (*i.e.* an element of the sequence $\mathcal{S}_{\mathcal{B}}$) contains some node labelled by $a : F$. We recall moreover that any mapping θ_i (guiding the construction of $\mathcal{N}_i$ as defined above) is injective, hence its inverse θ_i^- is defined.

Lemma 8. *Let $\mathcal{B}$ be a complete and open branch. For each set $\mathcal{N}_i$ belonging to $\mathcal{S}_{\mathcal{B}}$:*

1. *If $i > 0$ and d is a nominal occurring in $\mathcal{N}_{i-1}$, then no new node n^i added at stage i has a label of the form $d : p$ for $p \in \mathsf{PROP}$, or $d : \Box G$. As a consequence, if two nominals are compatible in $\mathcal{N}_i$, for any i, they stay compatible in $\mathcal{N}_{i+1}$ (and in $\mathcal{N}_{\mathcal{B}}^{\infty}$).*

2. *If $i > 0$ and θ_i is the mapping used to extend $\mathcal{N}_{i-1}$ to $\mathcal{N}_i$, then for any nominal d, d and $\theta_i(d)$ are compatible in $\mathcal{N}_i$.*

3. *For every triple $(n, m, \pi) \in \mathbf{B}_i$ and for any nominal d, d and $\pi(d)$ are compatible in $\mathcal{N}_i$.*

Proof. The three items are proved simultaneously by induction on i. If $i = 0$, item 1 and 2 are vacuously true and the last one holds by the properties of the mappings (definition 3) and the fact that $\mathcal{N}_0$ contains every node $n \in \mathcal{B}$ labelled by $a : H$ with $H \in Cmp(\mathcal{B})$. So, let us assume that 1, 2 and 3 hold for $i - 1$.

1. Let $d : H$ be the label of any node n^i added at stage i, where either H is a propositional letter in PROP or it has the form $\Box G$ (in both cases H does not contain any non-top nominal). We prove that $d = b^i$, where b^i is the new nominal added at stage i. Since H contains only top nominals (Lemma 2), $\theta_i^-(H) = H$ and $d : H = \theta_i(d') : H$ for some d', *i.e.* $d' : H$ is the label of the node $n \in \mathcal{N}_0$. Since $(n^i)\, d : H$ has been added at stage i, no node in $\mathcal{N}_{i-1} \supseteq \mathcal{N}_0$ is labelled by $d : H$; therefore $d \neq d'$.

   Let θ_i be either $\{a_0/c_0, ..., a_k/c_k\}$ or $\{a_0/c_0, ..., a_k/c_k, b^i/b\}$. Since $d \neq d'$, $d' \in \{c_0, c_1, ..., c_k, b\}$. If it were $d' = c_j$ for some $j = 1, ..., k$, then the label of n^i would be $a_j : H$; consequently, since no node labelled by a formula already occurring in $\mathcal{N}_{i-1}$ can be added to $\mathcal{N}_i$, $a_j : H$ wouldn't be the label of any node in $\mathcal{N}_{i-1}$; but this is impossible, because, by the induction hypothesis (item 2) a_j and c_j are compatible in $\mathcal{N}_{i-1}$. Therefore, we are left with the only possibility that $d = b^i$.

2. Let θ_i be either $\{a_0/c_0, ..., a_k/c_k\}$ or $\{a_0/c_0, ..., a_k/c_k, b^i/b\}$, where $\{a_0/c_0, ..., a_k/c_k\}$ is the injective mapping π of some triple $(n, m, \pi) \in \mathbf{B}_{i-1}$. By the inductive hypothesis (item 3), a_j and c_j are compatible in $\mathcal{N}_{i-1}$, and, by item 1 (which has already been proved for $\mathcal{N}_i$), they stay compatible in $\mathcal{N}_i$. So, let us assume that $b \neq c_j$, so that $\theta_i(b) = b^i$ where b^i is the fresh nominal added at stage i, and let H be any propositional letter in PROP or formula of the form $\Box G$. We have:

   - If $b : H \in \mathcal{N}_i$ then $b^i : H \in \mathcal{N}_i$. In fact, if $b : H \in \mathcal{N}_i$ then $b : H \in \mathcal{N}_0$ (by item 1), therefore $b^i : H \in \mathcal{N}_i$ by construction (for all $b : H \in \mathcal{N}_0$, $\theta_i(b) : H \in \mathcal{N}_i$).

- If $b^i : H \in \mathcal{N}_i$ then $b : H \in \mathcal{N}_i$. In fact, for all $b^i : H \in \mathcal{N}_i$, $b : H \in \mathcal{N}_0 \subseteq \mathcal{N}_i$.

3. Let (n, m, π) be a new triple added to $\mathbf{B}_i$ at stage i. Then for some m' and σ, $(m', m, \sigma) \in \mathbf{B}_0$, and $\pi = (\theta_i \circ \sigma) |_{label(m)}$. Let d be any nominal. By the induction hypothesis, d and $\sigma(d)$, which are obviously compatible in $\mathcal{N}_0$, are compatible also in $\mathcal{N}_i$, by item 1 (which has already been proved for $\mathcal{N}_i$). By item 2 (already proved for $\mathcal{N}_i$, too), $\sigma(d)$ and $\theta_i(\sigma(d))$ are compatible in $\mathcal{N}_i$. Therefore d and $\pi(d)$ are compatible $\mathcal{N}_i$. □

The next important property of the sets $\mathcal{N}_i$ is a kind of saturation property.

Definition 12. *Let $\mathcal{B}$ be a complete and open branch, let $\mathcal{N}_i$ be an element of $\mathcal{S}_\mathcal{B}$, and let $\mathbf{B}_i$ be the corresponding blocking relation for $\mathcal{N}_i$. The set $\mathcal{N}_i$ is* pseudo-saturated *with respect to $\mathbf{B}_i$ if it satisfies the following properties:*

1. *no node in $\mathcal{N}_i$ is labelled by a formula of the form $a : \neg a$;*
2. *there are no pairs of nodes labelled by formulae of the form $a : p$ and $a : \neg p$, for $p \in \mathsf{PROP}$;*
3. *if any node in $\mathcal{N}_i$ is labelled by a formula of the form $a : b$ (where a and b are nominals), then $a = b$;*
4. *if $(n)\, a : F \wedge G \in \mathcal{N}_i$ then, for some m and k, $(m)\, a : F \in \mathcal{N}_i$ and $(k)\, a : G \in \mathcal{N}_i$;*
5. *if $(n)\, a : F \vee G \in \mathcal{N}_i$ then, for some m, either $(m)\, a : F \in \mathcal{N}_i$ or $(m)\, a : G \in \mathcal{N}_i$;*
6. *if $(n)\, a : b : F \in \mathcal{N}_i$ then, for some m, $(m)\, b : F \in \mathcal{N}_i$;*
7. *if $(n)\, a :\downarrow x.F \in \mathcal{N}_i$ then, for some m, $(m)\, a : F[a/x] \in \mathcal{N}_i$;*
8. *if $(n)\, a : \Diamond F \in \mathcal{N}_i$ and $\mathbf{B}_i$ contains no triple of the form (n, n', π), then, for some nominal b and some m and k, $(m)\, a : \Diamond b \in \mathcal{N}_i$ and $(k)\, b : F \in \mathcal{N}_i$;*
9. *if $(n)\, a : \Box F \in \mathcal{N}_i$ and $(m)\, a : \Diamond b \in \mathcal{N}_i$ then, for some k, $(k)\, b : F \in \mathcal{N}_i$.*

Lemma 9. *Let $\mathcal{B}$ be a complete and open branch, let $\mathcal{N}_i$ be an element of $\mathcal{S}_\mathcal{B}$, and let $\mathbf{B}_i$ the blocking relation for $\mathcal{N}_i$. Then $\mathcal{N}_i$ is pseudo-saturated with respect to $\mathbf{B}_i$.*

Proof. The proof is by induction on i. $\mathcal{N}_0$ is clearly pseudo-saturated with respect to $\mathbf{B}_0$ because $\mathcal{B}$ is a complete and open branch. In particular, item 9 of Definition 12 holds because, if $(m)\, a : \Diamond b \in \mathcal{N}_0$, then m is not a phantom in $\mathcal{B}$; and for any $(n)\, a : \Box G \in \mathcal{N}_0$, the pair (n, m) has been expanded in $\mathcal{B}$, so that for some k, $(k)\, b : G \in \mathcal{B}$. Since $m \rhd k$ (see Definition 8), k is not a phantom in $\mathcal{B}$, therefore $k \in \mathcal{N}_0$.

Let us now assume that $\mathcal{N}_{i-1}$ is pseudo-saturated. The pseudo-saturation property still holds for all nodes in $\mathcal{N}_{i-1}$. We show that also the newly added nodes satisfy pseudo-saturation.

1. If some node n^i in $\mathcal{N}_i$ is labelled by a formula of the form $a : \neg a$, then for some node $n \in \mathcal{N}_0$, $a : \neg a = \theta_i(label(n))$. Therefore $label(n) = \theta_i^-(a : \neg a) = c : \neg c$, for some nominal c, contradicting the fact that $\mathcal{N}_0$ is pseudo-saturated.

2. Let us assume that, for some $p \in \mathsf{PROP}$, $(n^j)\, a : p \in \mathcal{N}_i$ and $(m^k)\, a : \neg p \in \mathcal{N}_i$. By the induction hypothesis, n^j and m^k cannot be both in $\mathcal{N}_{i-1}$.

   So let us consider three cases:

   (a) $(n^j)\, a : p \in \mathcal{N}_{i-1}$ and $(m^k)\, a : \neg p \notin \mathcal{N}_{i-1}$, thus $k = i$ and $label(m^i) = \theta_i(label(m)) = \theta_i(c : \neg p)$ for some $(m)\, c : \neg p \in \mathcal{N}_0$ and nominal c such that $\theta_i(c) = a$. By item 2 of Lemma 8, a and c are compatible in $\mathcal{N}_i$, therefore $c : p \in \mathcal{N}_i$. Since c occurs in $\mathcal{N}_0$, by item 1 of Lemma 8, $c : p \in \mathcal{N}_0$, contradicting the fact that $\mathcal{N}_0$ is pseudo-saturated.

   (b) $(n^j)\, a : p \notin \mathcal{N}_{i-1}$ and $(m^k)\, a : \neg p \in \mathcal{N}_{i-1}$, thus $j = i$. By item 1 of Lemma 8, $a = b^i$ is the fresh nominal introduced at stage i, which does not occur in $\mathcal{N}_{i-1}$. So, it cannot be the case that $m^k \in \mathcal{N}_{i-1}$, *i.e.* this case is actually impossible.

   (c) Neither n^j nor m^k are in $\mathcal{N}_{i-1}$, and $j = k = i$. Since $label(n^i) = a : p$, by item 1 of Lemma 8, $a = b^i$ is the fresh nominal introduced at stage i. Therefore, for some nodes $n, m \in \mathcal{N}_0$, $label(n^i) = \theta_i(label(n))$ and $label(m^i) = \theta_i(label(m))$. Therefore, $label(n) = \theta_i^-(a : p) = \theta_i^-(a) : p$ and $label(m) = \theta_i^-(a : \neg p) = \theta_i^-(a) : \neg p$, contradicting the fact that $\mathcal{N}_0$ is pseudo-saturated.

3. Let n^i be a new node added at stage i and labelled by a formula of the form $a : b$, where where a and b are nominals. Therefore $a : b = \theta_i(c) : \theta_i(d)$ for some $c : d \in \mathcal{N}_0$. Since $\mathcal{N}_0$ is pseudo-saturated, $c = d$, therefore also $a = b$.

4. Let $(n^i)\, a : F \wedge G$ be a node newly added to $\mathcal{N}_i$, and let $c : F' \wedge G'$ be the label of the node $n \in \mathcal{N}_0$. By construction, $label(n^i) = a : F \wedge G = \theta_i(label(n)) = \theta_i(c) : \theta_i(F') \wedge \theta_i(G')$. Since $\mathcal{N}_0$ is pseudo-saturated, it contains nodes $(n_1)\, c : F'$ and $(n_2)\, c : G'$. Therefore $\mathcal{N}_i$ contains nodes labelled, respectively, by $\theta_i(label(n_1)) = \theta_i(c : F') = a : F$ and $\theta_i(label(n_2)) = \theta_i(c : G') = a : G$.

5. Let a node $n^i \in \mathcal{N}_i$ be labelled by $a : F \vee G$, and let $c : F' \vee G'$ be the label of the node $n \in \mathcal{N}_0$. Then $label(n^i) = \theta_i(label(n)) = \theta_i(c) : \theta_i(F') \vee \theta_i(G')$. Since $\mathcal{N}_0$ is pseudo-saturated, $\mathcal{N}_0$ contains either a node $(n_1)\, c : F'$ or a node $(n_2)\, c : G'$. As a consequence, either $\theta_i(c : F') = a : F$ or $\theta_i(c : G') = a : G$ occurs as a node label in $\mathcal{N}_i$.

6. Let a node $n^i \in \mathcal{N}_i$ be labelled by $a : b : F$, and let $c : d : F'$ be the label of the node $n \in \mathcal{N}_0$. Then $label(n^i) = \theta_i(label(n)) = \theta_i(c : d : F')$. Since $\mathcal{N}_0$ is pseudo saturated, it contains a node labelled by $d : F'$, therefore $\mathcal{N}_i$ contains a node labelled by $\theta_i(d : F') = b : F$.

7. Let us assume that $(n^i)\, a :\downarrow x.F \in \mathcal{N}_i$. Then $\theta_i^-(a :\downarrow x.F) = c :\downarrow x.G$ is the label of the node $n \in \mathcal{N}_0$, where $\theta_i(c) = a$ and $\theta_i(G) = F$. Since $\mathcal{N}_0$ is pseudo-saturated, it also contains a node labelled by $c : G[c/x]$. Therefore $\mathcal{N}_i$ contains a node labelled by $\theta_i(c : G[c/x]) = \theta_i(c) : \theta_i(G)[\theta_i(c)/x] = a : F[a/x]$.

8. Let $(n)\, a : \Diamond F \in \mathcal{N}_i$. If n has no witness in $\mathcal{N}_i$ (there is no nominal b and nodes $(m)\, a : \Diamond b \in \mathcal{N}_i$ and $(k)\, b : F \in \mathcal{N}_i$), then $\mathbf{B}_i$ contains a triple of the form (n, n', π), by the invariant 2 of the construction of $\mathcal{S}_{\mathcal{B}}$ defined in Section 5.1.

9. Let us assume that $(n)\, a : \Box F \in \mathcal{N}_i$ and $(m)\, a : \Diamond b \in \mathcal{N}_i$. By Lemma 2, F does not contain any non-top nominal, hence $\theta_i(F) = F$ for any i.

   We distinguish two cases:

   (a) $(n)\, a : \Box F \notin \mathcal{N}_{i-1}$. By item 1 of Lemma 8, then, $a = b^i$ is the new nominal introduced at stage i. Therefore, $\mathcal{N}_0$ contains nodes labelled by $\theta_i^-(b^i : \Box F) = \theta_i^-(b^i) : \Box F$ and $\theta_i^-(b^i : \Diamond b) = \theta_i^-(b^i) : \Diamond \theta_i^-(b)$. Since $\mathcal{N}_0$ is pseudo-saturated, $\theta_i^-(b) : F \in \mathcal{N}_0$, so that $\theta_i(\theta_i^-(b)) : F = b : F \in \mathcal{N}_i$.

   (b) $(n)\, a : \Box F \in \mathcal{N}_{i-1}$. If also $(m)\, a : \Diamond b \in \mathcal{N}_{i-1}$, then $b : F \in \mathcal{N}_{i-1} \subseteq \mathcal{N}_i$ by the induction hypothesis. Otherwise, $\theta_i^-(a) : \Diamond \theta_i^-(b) \in \mathcal{N}_0$.

   Let $d = \theta_i^-(a)$ and $d' = \theta_i^-(b)$. By item 2 of Lemma 8, a and d are compatible in $\mathcal{N}_i$, therefore $d : \Box F \in \mathcal{N}_i$. Moreover, since d occurs in $\mathcal{N}_0$, by item 1 of Lemma 8, $d : \Box F \in \mathcal{N}_0$. Since also $d : \Diamond d' \in \mathcal{N}_0$ and $\mathcal{N}_0$ is pseudo-saturated, $d' : F \in \mathcal{N}_0$, so that also $\theta_i(d') : F = b : F \in \mathcal{N}_i$. □

Now we have all what is needed to build a model of any complete and open branch $\mathcal{B}$, i.e an interpretation $\mathcal{M}$ such that for any node label $a : F \in \mathcal{B}$, F holds in the state denoted by a.

Lemma 10. *If $\mathcal{B}$ is a complete and open branch, then the possibly infinite set $\mathcal{N}_{\mathcal{B}}^{\infty}$ has a model.*

Proof. Let $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \mathcal{N}, \mathcal{I} \rangle$ be defined as follows: W is the set of all the nominals occurring in $\mathcal{N}_{\mathcal{B}}^{\infty}$ and, for $a, b \in W$: aRb if and only if $a : \Diamond b$ is the label of some node in $\mathcal{N}_{\mathcal{B}}^{\infty}$, $N(a) = a$ and, for any $p \in \mathsf{PROP}$, $p \in I(a)$ if and only if $a : p$ is the label of some node in $\mathcal{N}_{\mathcal{B}}^{\infty}$. Such an interpretation is well defined because of Lemma 9. Since $\bigcap_{i \in \mathbb{N}} \mathbf{B}_i = \emptyset$, every blockable node in $\mathcal{N}_{\mathcal{B}}^{\infty}$ has a witness. Exploiting this fact and Lemma 9, an easy induction on F shows that for any label $a : F$ of a node in $\mathcal{N}_{\mathcal{B}}^{\infty}$, $\mathcal{M}_a \models F$. □

Theorem 2 (Completeness)**.** *If there exists a complete and open branch for the formula F, then F is satisfiable.*

Proof. Let $\mathcal{B}$ be a complete and open branch in a tableau for the formula F, and let $a : F_0$ be its top formula. Since the equality rule may have been used during the construction of $\mathcal{B}$, $F_0 = F[c_1/a_1, ..., c_n/a_n]$, where $a_1, ..., a_n$ do not occur in $\mathcal{B}$. By Lemma 10, there exists a model of F_0. Such a model can easily be extended to a model of F, establishing that $I(a_i) = I(c_i)$ for $i = 1, ..., n$. □

6 Concluding Remarks

In this work a tableau calculus for $\mathsf{HL}(@, \downarrow)$ is defined, which is provably terminating and complete for formulae belonging to the fragment $\mathsf{HL}(@, \downarrow) \setminus \downarrow \Box$. A preliminary transformation of formulae into equisatisfiable ones turns the calculus into a decision procedure to test satisfiability of formulae in $\mathsf{HL}(@, \downarrow) \setminus \Box \downarrow \Box$.

The main features of the calculus can be summarized as follows. A tableau branch is a sequence of nodes, each of which is labelled by a satisfaction statement. Since nominal equalities are dealt with by means of substitution, different occurrences of the same formula may occur as labels of different nodes in a branch. The fact that, when two formulae become equal by the effect of substitution the corresponding nodes do not collapse, allows for the definition of a binary relation $\prec_{\mathcal{B}}$ on nodes which organizes them into a family of trees. Each tree in such a family has bounded width, and this is due to the fact that, when applying the two premisses $\Box$ rule, it is the minor premiss, labelled by a relational formula, which is taken to be the "main responsible" of the expansion.

The fact that each branch in any tree of nodes is finite is guaranteed by a blocking mechanism which forbids the application of the $\Diamond$ rule to a node n whenever it has already been applied to another node whose label is equal to the label of n, *modulo non-top nominal renaming*. Such a renaming is essential, because, in the presence of the binder, non-top nominals may occur in the body of any node label. The blocking mechanism is *anywhere* blocking, paired with indirect blocking, relying on the relation $\prec_{\mathcal{B}}$.

Such a mechanism differs from [4, 5], where calculi for hybrid logic with the global and converse modalities (and no binders) are defined. In fact, such calculi adopt ancestor blocking, where nominals (and not nodes) are blocked, and indirect blocking relies on a partial order on nominals (instead of nodes). Differently from [5], moreover, the calculus defined in this work does not require nominal deletion to ensure termination. This is due, again, to the fact that a branch is not a set of formulae.

Also the tableau system defined in [10] for hybrid logic with the difference and converse modalities makes use of ancestor blocking, relying on an ancestor relation among *nominals*. The blocking mechanism used for converse free formulae in the same work is different and more similar to ours. In fact, an existential formula, such as, for instance, $a : \Diamond F$, is blocked (independently from its outermost nominal a) whenever there exists a nominal b labelling both F and every formula G such that $a : \Box G$ is in the branch. However, the sub-calculus does not terminate unless applications of the $\Box$ rule are prioritized.

A tableau calculus testing satisfiability of formulae in the *constant-free clique guarded fragment* has been proposed in [8]. A restriction of the algorithm to the guarded fragment has been defined and implemented [9]. A tableau branch, in such calculi, is a tree of nodes, and the label of each node is a set of formulae. A node is directly blocked by a previously created node if, essentially, their labels are the same modulo constant renaming. Our comparison modulo renaming method was in fact originally inspired by [8, 9] (although there are some differences). A further contact point between such calculi and ours is anywhere blocking coupled with indirect blocking (which, in [8, 9] relies on the ancestor relation in the tree).

We are presently working at the next natural step, *i.e.* the extension of the calculus to the global and converse modalities, so as to obtain a tableau based decision procedure for the fragment $\mathsf{HL}(@, \downarrow, \mathsf{E}, \Diamond^-) \setminus \Box \downarrow \Box$.

References

[1] C. Areces, P. Blackburn, and M. Marx. A road-map on complexity for hybrid logics. In J. Flum and M. Rodríguez-Artalejo, editors, *Computer Science Logic*, volume 1683 of *LNCS*, pages 307–321. Springer, 1999.

[2] C. Areces and B. ten Cate. Hybrid logics. In P. Blackburn, F. Wolter, and J. van Benthem, editors, *Handbook of Modal Logics*, pages 821–868. Elsevier, 2007.

[3] P. Blackburn and J. Seligman. Hybrid languages. *Journal of Logic, Language and Information*, 4:251–272, 1995.

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

[5] S. Cerrito and M. Cialdea Mayer. Nominal substitution at work with the global and converse modalities. In L. Beklemishev, V. Goranko, and V. Shehtman, editors, *Advances in Modal Logic*, volume 8, pages 57–74. College Publications, 2010.

[6] H. Ganzinger and H. De Nivelle. A superposition decision procedure for the guarded fragment with equality. In *Proc. 14th Symposium on Logic in Computer Science*, pages 295–305. IEEE Computer Society Press, 1999.

[7] E. Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 64:1719–1742, 1998.

[8] C. Hirsch and S. Tobies. A tableau algorithm for the clique guarded fragment. In F. Wolter, H. Wansing, M. de Rijke, and M. Zakharyaschev, editors, *Advances in Modal Logic*, volume 3, pages 257–277. CSLI Publications, 2001.

[9] J. Hladik. Implementation and evaluation of a tableau algorithm for the guarded fragment. In U. Egly and C. G. Fermüller, editors, *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2002)*, volume 2381 of *LNAI*, pages 145–159. Springer, 2002.

[10] M. Kaminski and G. Smolka. Terminating tableau systems for hybrid logic with difference and converse. *Journal of Logic, Language and Information*, 18(4):437–464, 2009.

[11] B. ten Cate and M. Franceschet. Guarded fragments with constants. *Journal of Logic, Language and Information*, 14:281–288, 2005.

[12] B. ten Cate and M. Franceschet. On the complexity of hybrid logics with binders. In L. Ong, editor, *Proceedings of Computer Science Logic 2005*, volume 3634 of *LNCS*, pages 339–354. Springer, 2005.