# Terminating Tableaux for Hybrid Logic with the Difference Modality and Converse

Mark Kaminski and Gert Smolka

Programming Systems Lab, Saarland University,
Campus E1 3, 66123 Saarbrücken, Germany
{kaminski,smolka}@ps.uni-sb.de

**Abstract.** We present the first terminating tableau calculus for basic hybrid logic with the difference modality and converse modalities. The language under consideration is basic multi-modal logic extended with nominals, the satisfaction operator, converse, global and difference modalities. All of the constructs are handled natively.
To obtain termination, we extend chain-based blocking for logics with converse by a complete treatment of difference.
Completeness of our calculus is shown via a model existence theorem that refines previous constructions by distinguishing between modal and equational state equivalence.

**Key words:** modal and hybrid logics, difference modality, converse, tableau systems, decision procedures

## 1 Introduction

Modal logic with the difference modality $Dp = \lambda x.\, \exists y.\, x{\neq}y \wedge py$ is an expressive language [1, 2]. It can express the global modality $Ep = p \dot{\vee} Dp$ and nominals $!p = E(p \dot{\wedge} \dot{\neg}(Dp))$. Gargov and Goranko [3] show that basic modal logic with $D$ is equivalent with respect to modal definability to basic hybrid logic [2, 4] with $E$ (see also [5–8]).

Tableaux for modal logic with $D$ are not well-understood. In a recent handbook chapter on modal proof theory [9], an unsound tableau calculus for basic modal logic with $D$ is given.[1] A sound and complete tableau calculus for basic modal logic with $D$ is given by Balbiani and Demri [1]. Unfortunately, Balbiani and Demri's calculus does not yield a decision procedure as it does not terminate on all inputs.

This paper presents a terminating prefixed tableau calculus for basic hybrid logic with $D$ and converse. While it is possible to express the satisfaction operator @ and $E$ in terms of $D$, it is more efficient to let the decision procedure handle satisfaction and global modalities natively. Hence, we allow @ and $E$ as additional constructs in our language and extend the calculus to deal with them directly. So, the input language for our calculus is precisely characterized as basic

[1] The formula $\Box P \vee D\neg P$ is invalid but provable by the rules in [9].

multi-modal logic extended with nominals, the satisfaction operator, converse, global and difference modalities.

The first tableau-based decision procedure for a modal language extended with $D$ as an additional operator was given in [10]. The blocking technique used there to ensure termination, called pattern-based blocking, is different from the traditional chain-based techniques [11–14] in that it does not exploit any information about the order in which prefixes are introduced to a tableau branch. In the presence of converse, however, pattern-based blocking as proposed in [10] is inherently incomplete.

Termination of the present calculus is obtained by chain-based blocking. Chain-based blocking was initially developed to deal with transitive modalities [11–13] and subsequently extended to logics with converse [15, 16] and nominals [14, 17]. As we show, the interaction between converse and $D$ is similar to the interaction between converse and transitive modalities, and can be handled by adapting the techniques in [15, 16] to account for the additional generative power of $D$.

Bolander and Blackburn [14] propose a different extension of chain-based blocking to global modalities and converse, blocking $E$ by the same mechanism as diamond modalities. We propose an alternative treatment of global modalities. Besides, our approach differs from that of [14] in the model construction techniques employed to prove completeness of our calculus. As Bolander and Blackburn's approach does not cover $D$, they employ traditional filtration arguments, constructing a model that identifies prefixes modulo modal equivalence. To capture the semantics of $D$, we construct a model that does not necessarily identify modally equivalent prefixes, while still respecting the stronger equational equivalence.

Unlike our approach, which is cumulative and relies solely on tree-like structures, Horrocks and Sattler [17] propose a tableau calculus for a nominal logic with global modalities and converse based on possibly cyclic graph structures and treat equational equivalence by destructive graph transformation during tableau construction. Their calculus does not cover $D$ but handles qualified number restrictions [18].

To treat $D$ in a sound and complete way, the calculus by Balbiani and Demri [1] employs a computationally expensive cut rule. To avoid the general inefficiency coming with this rule, we follow [10] and integrate it into the rule for the dual of $D$. Thus the costs of the cut rule need only be paid if the dual of $D$ is used.

It is possible to obtain decision procedures for the language under consideration by means of satisfiability-preserving translations into simpler languages [3, 19, 20, 8] for which effective decision procedures are already available [21, 14, 10]. Our calculus yields the first effective decision procedure for modal logic with both $D$ and converse modalities that does not rely on transformations of the input into other languages.

The paper is organized as follows. We start by formulating hybrid logic in simple type theory. Next, we present the rules of our calculus. Then, we impose control on the rules and show that the restricted calculus is terminating. The terminating calculus is then shown complete by means of a model existence theorem.

## 2 Hybrid Logic with D and Converse

We represent modal logic in simple type theory, which gives us an expressive syntax and a solid foundation. The basic idea of the representation goes back to Gallin [22] and can also be found in Gamut [23] (Sect. 5.8, two-sorted type theory). The representation of boxes and diamonds as higher-order constants appears in [24, 25]. Since the type-theoretic representation formalizes the semantics of modal logic at the object level, one can prove meta- and object-level theorems of modal logic with a higher-order theorem prover [26].

We start with two base types B and S. The interpretation of B is fixed and consists of two truth values. The interpretation of S is a nonempty set whose elements are called worlds or states. Given two types $\sigma$ and $\tau$, the *functional type* $\sigma\tau$ is interpreted as the set of all total functions from the interpretation of $\sigma$ to the interpretation of $\tau$. We write $\sigma_1\sigma_2\sigma_3$ for $\sigma_1(\sigma_2\sigma_3)$.

We employ three kinds of variables: *Nominal variables* $x$, $y$, $z$ of type S, *propositional variables* $p$, $q$ of type SB, and *relational variables* $r$ of type SSB. Nominal variables are called *nominals* for short. We use the logical constants

$$\begin{aligned} \bot, \top &: \mathrm{B} & \qquad \dot{=} &: \mathrm{SSB} \\ \neg &: \mathrm{BB} & \exists, \forall &: (\mathrm{SB})\mathrm{B} \\ \vee, \wedge, \rightarrow &: \mathrm{BBB} \end{aligned}$$

Terms are defined as usual. We write $st$ for applications, $\lambda x.s$ for abstractions, and $s_1s_2s_3$ for $(s_1s_2)s_3$. We also use infix notation, e.g., $s \wedge t$ for $(\wedge)st$.

Terms of type B are called *formulas*. We employ some common notational conventions: $\exists x.s$ for $\exists(\lambda x.s)$, $\forall x.s$ for $\forall(\lambda x.s)$, and $x \dot{\neq} y$ for $\neg(x \dot{=} y)$.

The formulas of modal logic can be either translated to type-theoretic formulas (as in [22, 23, 25, 27]) or directly represented as terms of type SB (as in [24, 26, 10]). Here we use the latter approach, which is more elegant since it models modal syntax directly as higher-order syntax. To do so, we need lifted versions of the Boolean connectives, which are defined as follows:

$$\begin{aligned} \dot{\neg} px &= \neg(px) & \qquad \dot{\neg} &: (\mathrm{SB})\mathrm{SB} \\ (p \mathbin{\dot{\wedge}} q)x &= px \wedge qx & \dot{\wedge} &: (\mathrm{SB})(\mathrm{SB})\mathrm{SB} \\ (p \mathbin{\dot{\vee}} q)x &= px \vee qx & \dot{\vee} &: (\mathrm{SB})(\mathrm{SB})\mathrm{SB} \end{aligned}$$

We can now write terms like $p \dot\wedge \dot\neg q$, which represent modal formulas. Here are the definitions of the remaining *modal constants* we will use:

$$
\begin{array}{rll}
r^- xy = ryx & & \_^- : (\mathrm{SSB})\mathrm{SSB} \\
\langle r\rangle px = \exists y.\, rxy \wedge py & & \langle\_\rangle : (\mathrm{SSB})(\mathrm{SB})\mathrm{SB} \\
[r]px = \forall y.\, rxy \rightarrow py & & [\_] : (\mathrm{SSB})(\mathrm{SB})\mathrm{SB} \\
Epx = \exists p & & E : (\mathrm{SB})\mathrm{SB} \\
Apx = \forall p & & A : (\mathrm{SB})\mathrm{SB} \\
Dpx = \exists y.\, x \dot{\neq} y \wedge py & & D : (\mathrm{SB})\mathrm{SB} \\
\bar{D}px = \forall y.\, x \doteq y \vee py & & \bar{D} : (\mathrm{SB})\mathrm{SB} \\
\dot{x}y = x \doteq y & & \dot{\_} : \mathrm{SSB} \\
@xpy = px & & @ : \mathrm{S}(\mathrm{SB})\mathrm{SB}
\end{array}
$$

Applied to a relation $r$, the operator $\_^-$ yields the converse of $r$. This allows us to add converse to our language without introducing converse versions of the operators $\langle\_\rangle$ and $[\_]$. We call a term $t : \mathrm{SB}$ *modal* if it has the form

$$
\begin{array}{rcl}
\rho & ::= & r \mid r^- \\
t & ::= & p \mid \dot\neg t \mid t \circ t \mid \mu\rho t \mid \nu t \mid \dot{x} \mid @xt
\end{array}
$$

where $\circ \in \{\dot\wedge, \dot\vee\}$, $\mu \in \{\langle\_\rangle, [\_]\}$ and $\nu \in \{E, A, D, \bar{D}\}$.

A *modal interpretation* $\mathfrak{M}$ is an interpretation of simple type theory that interprets B as the set $\{0, 1\}$, $\bot$ as 0 (i.e., false), $\top$ as 1 (i.e., true), maps S to a non-empty set, gives the logical constants $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\exists$, $\forall$, $\doteq$ their usual meaning, and satisfies the equations defining the modal constants $\dot\neg$, $\dot\wedge$, $\dot\vee$, $\_^-$, $\langle\_\rangle$, $[\_]$, $E$, $A$, $D$, $\bar{D}$, $\dot{\_}$, and @. Whenever $\mathfrak{M}t = 1$, we say that $\mathfrak{M}$ *satisfies* $t$, or that $\mathfrak{M}$ is a *model* of $t$. A formula is called *satisfiable* if it has a satisfying modal interpretation.

We now give some additional syntactic definitions that are needed for the rest of the paper. A modal term $s : \mathrm{SB}$ is called *normal* if it is in negation-normal form, that is, has the form

$$
s \;::=\; p \mid \dot\neg p \mid s \circ s \mid \mu\rho s \mid \nu s \mid \dot{x} \mid \dot\neg\dot{x} \mid @xs
$$

where $\circ \in \{\dot\wedge, \dot\vee\}$, $\mu \in \{\langle\_\rangle, [\_]\}$ and $\nu \in \{E, A, D, \bar{D}\}$. A formula $s$ is called *normal* if it has the form $tx$ where $t$ is a normal modal term. Formulas of the form $rxy$ or $r^- xy$ are called *accessibility formulas* or *edges*.

Given a term $t$, we write $\mathcal{N}t$ for the set of nominals that occur in $t$. The notation is extended to sets of terms in the natural way: $\mathcal{N}X := \bigcup\{\mathcal{N}t \mid t \in X\}$.

## 3 Tableau Rules

Our tableaux are constructed in the usual way from a finite non-empty set of *initial* normal formulas by the rules in Fig. 1. The rules may extend tableau branches by formulas $s$ of the form

$$
s \;::=\; x \doteq y \mid x \dot{\neq} y \mid \rho xy \mid tx
$$

where $t$ is a normal modal term. Single tableau branches are referred to by the meta-variables $\Gamma$ and $\Delta$. We allow no multiple occurrences of identical formulas on a single branch. Nominals $x$ occurring in normal formulas $sx$ are used to reference individual states, analogously to prefixes as used by related calculi [28, 14] and, similarly, prefixed calculi for nominal-free logics [9], with the important difference that in our case prefixes are part of the object language. We use edges to represent assertions about accessibility relations and equations for state equality or inequality constraints.

Given a branch $\Gamma$, we use $\sim_\Gamma$ to denote the equivalence closure of the relation $\{(x,y) \mid x \doteq y \in \Gamma\}$. If $x \sim_\Gamma y$, we call $x$ and $y$ *equationally equivalent* on $\Gamma$.

It is easy to verify that the rules in Fig. 1 are sound in the following sense.

**Proposition 1 (Soundness).** *Let $\Gamma$ be a tableau branch and $\Delta_1, \dots, \Delta_n$ be the extensions of $\Gamma$ obtained by a rule $\mathcal{R}$ from Fig. 1 ($n \in \{1,2\}$). Then $\Gamma$ is satisfiable if and only if there is some $i \in \{1,\dots,n\}$ such that $\Delta_i$ is satisfiable.*

Unlike [28, 14] but similarly to [15, 17], we use signed edges of the form $rxy$ and $r^-xy$. We define $\tilde{r} := r^-$ and $\widetilde{r^-} := r$. Semantically, $rxy$ is considered identical to $r^-yx$. But the former formula additionally records that $y$ was added to the branch after $x$, while the latter formula implies the converse. This way, we have an explicit representation of all the chronological information that will be necessary in Sect. 4 to impose a terminating control on the rules.

As all the relevant chronological information is contained in the edges, we can ignore the vertical structure of tableau branches and see them as sets of formulas, which may be subject to the usual set predicates and operators. For instance, we may write $s \in \Gamma$ to denote that $s$ occurs on $\Gamma$, and $\Gamma - \Delta$ for the set of formulas that occur on $\Gamma$ but not on $\Delta$. The notation $\mathcal{N}\Gamma$ is defined in the obvious way.

$$\mathcal{R}_{\dot\wedge}\ \frac{(s \mathbin{\dot\wedge} t)x}{sx,\ tx} \qquad \mathcal{R}_{\dot\vee}\ \frac{(s \mathbin{\dot\vee} t)x}{sx \ \mid\ tx} \qquad \mathcal{R}_{\Diamond}\ \frac{\langle\rho\rangle tx}{\rho xy,\ ty}\ y \notin \mathcal{N}\Gamma \qquad \mathcal{R}_{\Box}\ \frac{[\rho]tx \qquad \rho xy}{ty}$$

$$\mathcal{R}_{\tilde\Box}\ \frac{[\rho]tx \qquad \tilde\rho yx}{ty} \qquad \mathcal{R}_{E}\ \frac{Etx}{ty}\ y \notin \mathcal{N}\Gamma \qquad \mathcal{R}_{A}\ \frac{Atx}{ty}\ y \in \mathcal{N}\Gamma$$

$$\mathcal{R}_{\doteq}\ \frac{sx}{sy}\ x \sim_\Gamma y,\ s \text{ modal} \qquad \mathcal{R}_{N}\ \frac{\dot{x}y}{x \doteq y} \qquad \mathcal{R}_{\bar N}\ \frac{\dot\neg \dot{x}y}{x \mathrel{\dot\neq} y} \qquad \mathcal{R}_{@}\ \frac{@ytx}{ty}$$

$$\mathcal{R}_{D}\ \frac{Dtx}{x \mathrel{\dot\neq} y,\ ty}\ y \notin \mathcal{N}\Gamma \qquad \mathcal{R}_{\bar D}\ \frac{\bar{D}tx}{x \doteq y \ \mid\ ty}\ y \in \mathcal{N}\Gamma$$

$\Gamma$ is the tableau branch to which a rule is applied.

**Fig. 1.** Tableau Rules

We call a branch $\Gamma$ *closed* if there is some $p$, $x$ and $y$ such that $\Gamma$ contains either both $px$ and $\dot\neg px$ or a disequation $x{\not=}y$ where $x \sim_\Gamma y$. Otherwise, $\Gamma$ is called *open*. A tableau is called closed if all of its branches are closed, and open otherwise. To prove a modal term $s$ valid, one computes the negation-normal form $t$ of $\dot\neg s$, selects a nominal $x \notin \mathcal{N}t$, and constructs a closed tableau for $tx$.

## 4 Control

It is easy to see that our tableau rules do not terminate without additional restrictions on their applicability. Figure 2 shows a possible non-terminating derivation. So, to achieve termination, we need to impose on our rules a terminating control.

| | |
|---|---|
| $A(\langle r\rangle p)x$ | |
| $\langle r\rangle px$ | $\mathcal{R}_A$ |
| $rxy,\ py$ | $\mathcal{R}_\Diamond$ |
| $\langle r\rangle py$ | $\mathcal{R}_A$ |
| $\ldots$ | |

**Fig. 2.** A Non-terminating Tableau Derivation

Every tableau branch $\Gamma$ can be seen as a graph with the vertices $\mathcal{N}\Gamma$ and the edges given by the relation $<_\Gamma := \{(x,y) \mid \exists \rho\colon\ \rho xy \in \Gamma\}$. The relations $<_\Gamma^+$ and $<_\Gamma^*$ are defined from $<_\Gamma$ as usual (transitive and reflexive transitive closure). We define $G_\Gamma := (\mathcal{N}\Gamma, <_\Gamma)$.

A modal term $s$ is said to *occur at a nominal $x$ on a tableau branch $\Gamma$* if $sx$ occurs on $\Gamma$. We define the *labeling* $\mathcal{L}_\Gamma x$ of a nominal $x$ on a branch $\Gamma$ to be set of all modal terms that occur at $x$ on $\Gamma$. Two nominals $x, y$ are called *modally equivalent* on a branch $\Gamma$ if $\mathcal{L}_\Gamma x = \mathcal{L}_\Gamma y$. The function $\mathcal{L}_\Gamma$ defines a vertex labeling of $G_\Gamma$ with sets of modal terms. We say a nominal $x$ is a *root* of $G_\Gamma$ if $x$ has no predecessor in $<_\Gamma$, and write $\mathrm{Root}\,\Gamma$ for the set of all roots of $G_\Gamma$.

The graph $G_\Gamma$ should *not* be understood as a partial model of $\Gamma$. So, the connection between $<_\Gamma$ and the transition relations in possible models of $\Gamma$ is relatively loose. In particular, our tableau algorithm will always keep $<_\Gamma$ acyclic while actual models of $\Gamma$ may contain cycles.

Achieving termination is easy once we can give an upper bound on the number of vertices in $G_\Gamma$. In particular, we would like to be able to bound the maximal length of chains $x_1 <_\Gamma \ldots <_\Gamma x_n$. To do so, we want to avoid extending such chains if they are repeating, i.e., contain two distinct nominals with the same labeling. This motivates the following definition: A nominal $x$ is called *active* on a branch $\Gamma$ if there are no two distinct nominals $y, z <^*_\Gamma x$ such that $\mathcal{L}_\Gamma y = \mathcal{L}_\Gamma z$. Otherwise, $x$ is called *inactive*.

We say a formula $s$ is *expanded* on a branch $\Gamma$ if one of the following *expandedness conditions* holds:

($\mathcal{E}_\wedge$) $s = (t_1 \dot\wedge t_2)x$ and $t_1x, t_2x \in \Gamma$
($\mathcal{E}_\vee$) $s = (t_1 \dot\vee t_2)x$ and $t_1x \in \Gamma$ or $t_2x \in \Gamma$
($\mathcal{E}_\Diamond$) $s = \langle\rho\rangle tx$ and there is some $y$ such that $ty \in \Gamma$ and either $\rho xy \in \Gamma$ or $\tilde\rho yx \in \Gamma$
($\mathcal{E}_\Box$) $s = [\rho]tx$ and, for every $y$ such that $\rho xy \in \Gamma$ or $\tilde\rho yx \in \Gamma$, it holds $ty \in \Gamma$
($\mathcal{E}_E$) $s = Etx$ and there is some $y \in \mathrm{Root}\,\Gamma$ such that $ty \in \Gamma$
($\mathcal{E}_A$) $s = Atx$ and, for every $y \in \mathcal{N}\Gamma$, it holds $ty \in \Gamma$
($\mathcal{E}_{\dot=}$) $s = x\dot=y$ and $\mathcal{L}_\Gamma x = \mathcal{L}_\Gamma y$
($\mathcal{E}_N$) $s = \dot y x$ and $y\dot=x \in \Gamma$
($\mathcal{E}_{\bar N}$) $s = \dot\neg\dot y x$ and $y\dot\neq x \in \Gamma$
($\mathcal{E}_@$) $s = @ytx$ and $ty \in \Gamma$
($\mathcal{E}_D$) $s = Dtx$ and there is some $y \in \mathrm{Root}\,\Gamma$ such that $y \not\sim_\Gamma x$ and $ty \in \Gamma$
($\mathcal{E}_{\bar D}$) $s = \bar Dtx$ and, for every $y \in \mathcal{N}\Gamma$, either $x \sim_\Gamma y$ or $ty \in \Gamma$

Note that there are no expandedness conditions for formulas of the form $px$, $\dot\neg px$ and $x\dot\neq y$.

We restrict the applicability of our tableau rules by two conditions.

($\mathcal{C}_1$) A rule is applicable to a formula $s \in \Gamma$ only if $\Gamma$ is open, $s$ is not expanded on $\Gamma$, and if the rule application results in a *proper extension* of $\Gamma$, i.e., extends $\Gamma$ by at least one formula that does not already occur on $\Gamma$.
($\mathcal{C}_2$) A rule is applicable to a formula of the form $\langle\rho\rangle tx$ on $\Gamma$ only if $x$ is active on $\Gamma$.

Note that $\mathcal{C}_1$ applies to all formulas, including diamonds, while $\mathcal{C}_2$ applies to diamond formulas only.

Except possibly for the cases $\mathcal{E}_E$ and $\mathcal{E}_D$, the condition $\mathcal{C}_1$ is intuitive. Indeed, similar conditions are often assumed implicitly when formulating tableau systems. The restriction $\mathcal{C}_2$ is a chain-based blocking condition as in [15, 16].

Incidentally, $\mathcal{E}_\Diamond$ has a well-known analog in tableaux for classical first-order logic. There, the applicability of the existential rule $\delta$ can be restricted to once per formula. In a somewhat less obvious way, $\mathcal{E}_E$ and $\mathcal{E}_D$ also relate to this restriction. More details are provided later.

We are going to show that our calculus with the two applicability restrictions is complete and terminating, thus yielding a decision procedure for hybrid logic with $D$ and converse. If a branch cannot be extended by any tableau rules, we call it *maximal*. Assuming that our calculus terminates, its completeness is proven by showing that an open and maximal tableau branch always exhibits a model of its initial formulas.

In the cases $\mathcal{E}_E$ and $\mathcal{E}_D$, it may seem unclear why we want the witness of $s$ (i.e., the nominal $y$ such that $ty \in \Gamma$) to be a root of $G_\Gamma$. One may consider taking the following weaker versions of $\mathcal{E}_E$ and $\mathcal{E}_D$:

($\mathcal{E}'_E$) $Etx$ is expanded if there is some $y$ such that $ty \in \Gamma$.
($\mathcal{E}'_D$) $Dtx$ is expanded if there is some $y$ such that $x \not\sim_\Gamma y$ and $ty \in \Gamma$.

It turns out, however, that if we do so, the interaction of $\mathcal{C}_1$ with $\mathcal{C}_2$ will render our calculus incomplete. Figure 3 shows an open branch for the unsatisfiable set $\{A(\langle r\rangle p)x, A(\langle r\rangle\dot\bot \mathbin{\dot\vee} E(\langle r\rangle\dot\bot))x\}$, where $\dot\bot := q \mathbin{\dot\wedge} \dot\neg q$, which becomes maximal if we weaken $\mathcal{E}_E$ to $\mathcal{E}'_E$. An example for $\mathcal{E}'_D$ looks analogously.

| | |
|---|---|
| $A(\langle r\rangle p)x,\ A(\langle r\rangle\dot\bot \mathbin{\dot\vee} E(\langle r\rangle\dot\bot))x$ | |
| $\langle r\rangle px$ | $\mathcal{R}_A$ |
| $(\langle r\rangle\dot\bot \mathbin{\dot\vee} E(\langle r\rangle\dot\bot))x$ | $\mathcal{R}_A$ |
| $E(\langle r\rangle\dot\bot)x$ | $\mathcal{R}_{\dot\vee}$ |
| $rxy,\ py$ | $\mathcal{R}_\Diamond$ |
| $\langle r\rangle py$ | $\mathcal{R}_A$ |
| $(\langle r\rangle\dot\bot \mathbin{\dot\vee} E(\langle r\rangle\dot\bot))y$ | $\mathcal{R}_A$ |
| $E(\langle r\rangle\dot\bot)y$ | $\mathcal{R}_{\dot\vee}$ |
| $ryz,\ pz$ | $\mathcal{R}_\Diamond$ |
| $\langle r\rangle pz$ | $\mathcal{R}_A$ |
| $rzu,\ pu$ | $\mathcal{R}_\Diamond$ |
| $\langle r\rangle pu$ | $\mathcal{R}_A$ |
| $(\langle r\rangle\dot\bot \mathbin{\dot\vee} E(\langle r\rangle\dot\bot))u$ | $\mathcal{R}_A$ |
| $\langle r\rangle\dot\bot u$ | $\mathcal{R}_{\dot\vee}$ |
| $(\langle r\rangle\dot\bot \mathbin{\dot\vee} E(\langle r\rangle\dot\bot))z$ | $\mathcal{R}_A$ |
| $E(\langle r\rangle\dot\bot)z$ | $\mathcal{R}_{\dot\vee}$ |

**Fig. 3.** A Maximal Tableau Branch with the Expandedness Condition $\mathcal{E}'_E$

Another variant of $\mathcal{E}_D$ that we might consider corresponds more closely to the tableau rule for $D$:

($\mathcal{E}''_D$) $Dtx$ is expanded if there is some $y$ such that $x \dot{\neq} y, ty \in \Gamma$.

Here, it is termination that is no longer guaranteed, as shown in Fig. 4.

## 5 Termination

We will now show that every tableau derivation is finite. Since the two branching rules $\mathcal{R}_{\dot\vee}$ and $\mathcal{R}_{\bar D}$ are at most binary, by König's lemma it suffices to show that the length of the individual branches is bounded.

| | |
|---|---|
| $A(D(D(Dp)))x$ | |
| $D(D(Dp))x$ | $\mathcal{R}_A$ |
| $x \dot{\neq} y,\ D(Dp)y$ | $\mathcal{R}_D$ |
| $y \dot{\neq} z,\ Dpz$ | $\mathcal{R}_D$ |
| $z \dot{\neq} u,\ pu$ | $\mathcal{R}_D$ |
| $D(D(Dp))u$ | $\mathcal{R}_A$ |
| $\ldots$ | |

**Fig. 4.** A Non-terminating Tableau Derivation with the Expandedness Condition $\mathcal{E}''_D$

Since every rule application extends a branch only by formulas that do not yet occur on the branch, the length of a branch $\Gamma$ coincides with the number of formulas on $\Gamma$. First, let us show that this number is bounded by a function in the number of nominals on $\Gamma$. Then, we will show that this number is itself bounded from above, completing the termination proof.

We write $\Gamma \rightarrow \Delta$ to denote that the branch $\Delta$ is an extension of a branch $\Gamma$ obtained by a single rule application. The notations $\Gamma \rightarrow^+ \Delta$ and $\Gamma \rightarrow^* \Delta$ are then defined in the obvious way. We write $\mathrm{Mod}\,\Gamma$ for the set of all modal terms occurring on $\Gamma$, possibly as subterms, and $\mathrm{Rel}\,\Gamma$ for the set of all relational variables that occur on $\Gamma$.

Crucial for our termination argument is the fact that our rules cannot introduce to the tableau any modal terms that do not already occur as subterms of the initial formulas.

**Proposition 2 (Subterm Property).** *If* $\Gamma \rightarrow^* \Delta$*, then* $\mathrm{Mod}\,\Gamma = \mathrm{Mod}\,\Gamma$*.*

For every pair of nominals $x, y$ and every relation $r$, a branch $\Gamma$ may contain edges $rxy$ and $r^- xy$, equations $x \dot{=} y$, disequations $x \dot{\neq} y$ and, for every term $s \in \mathrm{Mod}\,\Gamma$, a formula $sx$. Hence, the size of $\Gamma$ is bounded by $2|\mathrm{Rel}\,\Gamma| \cdot |\mathcal{N}\Gamma|^2 + 2|\mathcal{N}\Gamma|^2 + |\mathrm{Mod}\,\Gamma| \cdot |\mathcal{N}\Gamma|$. By Proposition 2, we know that $|\mathrm{Mod}\,\Gamma|$ and $|\mathrm{Rel}\,\Gamma|$ depend only on the initial formulas of the tableau.

So, it suffices to show that $|\mathcal{N}\Gamma|$ is bounded. We do so by showing that $G_\Gamma$ is a finite forest of a size bounded by some function in the initial branch $\Gamma_0$. Looking at how $\Gamma$ is constructed, it is easy to see that $G_\Gamma$ is a well-founded forest, so it remains to show that:

1. Every tree in $G_\Gamma$ has bounded outdegree.
2. Every tree in $G_\Gamma$ has bounded depth.
3. $G_\Gamma$ has a bounded number of roots.

The first bound is obtained by observing that edges are only added by the rule $\mathcal{R}_\Diamond$. It is easy to see that once $\mathcal{R}_\Diamond$ is applied to some formula $s$, $s$ will be expanded on all extensions of the resulting branch. Hence, the outdegree of a nominal $x$ is bounded by the number of distinct terms $\langle\rho\rangle t$ that occur at $x$, which, in its turn, is bounded by $|\mathrm{Mod}\,\Gamma_0|$.

The bound on the depth of the trees in $G_\Gamma$ is $2^{|\mathrm{Mod}\,\Gamma_0|}+1$, which easily follows from the fact that, by the Subterm Property and the pigeonhole principle, every sequence $x_1 <_\Gamma \dots <_\Gamma x_{2^{|\mathrm{Mod}\,\Gamma_0|}+1}$ contains at least two distinct but modally equivalent nominals.

Now to the the number of roots in $G_\Gamma$. The applicability condition $\mathcal{C}_1$ enforces that the number of distinct formulas on a branch is strictly increased by every rule application.

**Proposition 3.** *If $\Gamma \to^+ \Delta$, then $\Gamma \subsetneq \Delta$.*

Note that since our tableaux are constructed starting from normal formulas, $<_{\Gamma_0}$ is always empty. Hence, since $\Gamma_0$ is non-empty, $\mathrm{Root}\,\Gamma$ contains at least one nominal. Moreover, whenever a branch $\Gamma$ is extended by a formula $\rho xy$, we require that $y \notin \mathcal{N}\Gamma$. Therefore, once a nominal is a root of $\Gamma$, it will remain a root for every extension of $\Gamma$.

**Proposition 4.** *If $\Gamma \to^* \Delta$, then $\mathrm{Root}\,\Gamma \subseteq \mathrm{Root}\,\Delta$.*

Since there are only two rules that can introduce new roots to $G_\Gamma$, namely $\mathcal{R}_E$ and $\mathcal{R}_D$, it suffices to show that the number of their applications in any derivation is bounded from above by a function in the initial branch $\Gamma_0$. The bound for $\mathcal{R}_E$ is given by $\mathcal{B}_E\Gamma_0$, and the bound for $\mathcal{R}_D$ by $\mathcal{B}_D\Gamma_0$, where $\mathcal{B}_E$ and $\mathcal{B}_D$ are defined as follows.

$$\mathcal{B}_E\Gamma \;:=\; |\mathrm{Mod}\,\Gamma - \{\, s \mid \exists x \in \mathrm{Root}\,\Gamma \colon sx \in \Gamma \,\}|$$

The intuition behind $\mathcal{B}_E\Gamma$ is that $\mathcal{R}_E$ can only be applied once per modal term, independently of the nominal at which the term occurs. By Propositions 2, 3 and 4, $\mathcal{B}_E\Gamma$ is decreased by every application of $\mathcal{R}_E$ and not increased by any of the other rules. The definition

$$\begin{aligned}\mathcal{B}_D\Gamma := {} & |\mathrm{Mod}\,\Gamma - \{s \mid \exists\, y \in \mathrm{Root}\,\Gamma \colon sy \in \Gamma\}| \\ & + |\mathrm{Mod}\,\Gamma - \{s \mid \exists\, x \in \mathcal{N}\Gamma\; \exists\, y,z \in \mathrm{Root}\,\Gamma \colon x \sim_\Gamma y \text{ and } \{sy, x \dot{\neq} z, sz\} \subseteq \Gamma\}|\end{aligned}$$

follows the same idea, with the intuition here being that $\mathcal{R}_D$ is applicable at most twice per modal term. One can verify that $\mathcal{B}_D\Gamma$ is decreased by $\mathcal{R}_D$ and not increased by any of the other rules. That the second argument of the sum is needed can be seen with the branch $\{Dsx,\ x \dot{\neq} y,\ sy, Dsy\}$, where $y$ is a root and $Dsy$ is not expanded. To see that $\mathcal{R}_D$ is not applicable to a formula $Dsu \in \Gamma$ once, for some $x \in \mathcal{N}\Gamma$ and $y, z \in \mathrm{Root}\,\Gamma$, it holds $x \sim_\Gamma y$ and $\{sy,\ x \dot{\neq} z,\ sz\} \subseteq \Gamma$, observe that $Dsx$ is expanded unless $x \sim_\Gamma y \sim_\Gamma z \sim_\Gamma u$, in which case $\Gamma$ is closed and hence maximal.

## 6 Model Existence

To prove our calculus complete, it remains to show that every open maximal extension $\Gamma$ of an initial branch $\Gamma_0$ exhibits a model $\mathfrak{M}$ of $\Gamma_0$. Without converse modalities, we can construct $\mathfrak{M}$ such that it satisfies not only $\Gamma_0$ but all formulas on $\Gamma$ [28, 14, 10]. With converse, however, it seems easier to construct a model only for a distinguished subset $X$ of $\Gamma$ that still contains $\Gamma_0$. It is known [15, 14] that the set of formulas occurring at nominals active on $\Gamma$ is a suitable candidate for $X$.

The model construction by Bolander and Blackburn [14] deals with equational equivalence of nominals by identifying nominals up to modal equivalence (this approach is commonly known as filtration). Two nominals $x$ and $y$ are mapped to the same state if $\mathcal{L}_\Gamma x = \mathcal{L}_\Gamma y$. This suffices because on saturated branches equational equivalence implies modal equivalence. However, the approach is no longer appropriate once we extend our language by $D$. Look at the branch $\Gamma := \{Dpx, px, Dpy, py\}$. A model of $\Gamma$ needs at least two different states, both of which may satisfy the same set of formulas. To avoid this problem, we base our model construction not on modal equivalence but directly on equational equivalence as defined by the relation $\sim_\Gamma$.

We proceed in several steps. Starting with a branch $\Gamma$, we apply to it a substitution $\varphi$ eliminating syntactically distinct nominals that are equivalent modulo $\sim_\Gamma$. Then, we construct a model $\mathfrak{M}$ of a distinguished subset $\varphi X$ of $\varphi\Gamma$ such that $X$ contains $\Gamma_0$. Finally, we show how to extend $\mathfrak{M}$ to a model of $X$.

A *nominal substitution* $\varphi$ is a function $\mathit{Nom} \to \mathit{Nom}$, where $\mathit{Nom}$ is the set of all nominals. Since nominal substitutions are the only kind of substitutions we will look at, in the following we will refer to them simply as "substitutions". We write $\varphi s$ for the term obtained by replacing every nominal $x$ in $s$ by $\varphi x$. So, for instance, $\varphi((@x\dot{y})z) = (@x'\dot{y}')z'$ if $\varphi x = x'$, $\varphi y = y'$ and $\varphi z = z'$. Substitutions are extended to sets of terms in the intuitive way. Given a branch $\Gamma$, we call a substitution $\varphi$ a *normalizer* for $\Gamma$ if $\varphi x \sim_\Gamma x$ for all $x \in \mathcal{N}\Gamma$ and $\forall x, y \in \mathcal{N}\Gamma\colon\ \varphi x = \varphi y \iff x \sim_\Gamma y$. Note that, given an at most countable branch $\Gamma$, a normalizer $\varphi$ for $\Gamma$ can always be constructed by taking an arbitrary well-ordering $\prec$ of $\Gamma$ and setting $\varphi := \{\, (x, y) \in (\mathcal{N}\Gamma)^2 \mid y = \min_\prec \{\, z \in \mathcal{N}\Gamma \mid x \sim_\Gamma z \,\}\,\}$. Hence, normalizers exist for every branch $\Gamma$ of our calculus. They are not unique since neither are well-orderings of $\Gamma$.

**Lemma 1.** *Let $\Gamma$ be open and maximal. If $x \sim_\Gamma y$, then $\mathcal{L}_\Gamma x = \mathcal{L}_\Gamma y$.*

**Lemma 2.** *Let $\Gamma$ be open and maximal and $\varphi$ a normalizer for $\Gamma$. If $\mathcal{L}_\Gamma x = \mathcal{L}_\Gamma y$, then $\mathcal{L}_{\varphi\Gamma}(\varphi x) = \mathcal{L}_{\varphi\Gamma}(\varphi y)$.*

*Proof.* Clearly, $\mathcal{L}_{\varphi\Gamma}(\varphi z) = \bigcup_{u \sim_\Gamma z} \varphi(\mathcal{L}_\Gamma u)$. By Lemma 1, the latter is the same as $\varphi(\mathcal{L}_\Gamma z)$. So, $\mathcal{L}_{\varphi\Gamma}(\varphi x) = \varphi(\mathcal{L}_\Gamma x) = \varphi(\mathcal{L}_\Gamma y) = \mathcal{L}_{\varphi\Gamma}(\varphi y)$. □

A nominal $x$ is called *relevant* on $\Gamma$ if every $y$ such that $y <_\Gamma^+ x$ is active.

**Proposition 5.** *Every nominal that is active on a branch $\Gamma$ is relevant on $\Gamma$.*

**Proposition 6.** *If $x$ is active on $\Gamma$ and either $\rho xy \in \Gamma$ or $\rho yx \in \Gamma$, then $y$ is relevant on $\Gamma$.*

**Proposition 7.** *If $x$ is relevant on $\Gamma$, then there is some $y <_\Gamma^* x$ such that $y$ is active on $\Gamma$ and $\mathcal{L}_\Gamma y = \mathcal{L}_\Gamma x$.*

For the model construction, we want to eliminate all distinct nominals that are equationally equivalent. This will allow us to construct a term model of the initial branch in which syntactically distinct nominals denote distinct states. This is achieved by considering the image of a branch $\Gamma$ under a normalizer $\varphi$. Of course, applying $\varphi$ to $\Gamma$ will destroy the forest structure of $G_\Gamma$. The desired properties of $\varphi\Gamma$ can be formulated as follows.

A set $\Gamma$ of formulas is *saturated for a formula $sx \in \Gamma$ on a set $X \subseteq \mathcal{N}\Gamma$* if $\mathcal{N}(\mathrm{Mod}\,\Gamma) \subseteq X$ and one of the following *saturatedness conditions* holds:

($\mathcal{S}_\wedge$) $s = t_1 \dot\wedge t_2$ and $t_1x, t_2x \in \Gamma$
($\mathcal{S}_\vee$) $s = t_1 \dot\vee t_2$ and $t_1x \in \Gamma$ or $t_2x \in \Gamma$
($\mathcal{S}_\Diamond$) $s = [\rho]t$ and either $x \notin X$ or there is some $y \in \mathcal{N}\Gamma$ such that $ty \in \Gamma$, either $\rho xy \in \Gamma$ or $\tilde\rho yx \in \Gamma$, and $\mathcal{L}_\Gamma y = \mathcal{L}_\Gamma z$ for some $z \in X$
($\mathcal{S}_\Box$) $s = [\rho]t$ and, for every $y$ such that $\rho xy \in \Gamma$ or $\tilde\rho yx \in \Gamma$, it holds $ty \in \Gamma$
($\mathcal{S}_E$) $s = Et$ and there is some $y \in X$ such that $ty \in \Gamma$
($\mathcal{S}_A$) $s = At$ and, for every $y \in \mathcal{N}\Gamma$, it holds $ty \in \Gamma$
($\mathcal{S}_N$) $s = \dot y$ and $y = x$
($\mathcal{S}_{\bar N}$) $s = \dot\neg\dot y$ and $y \dot{\neq} x \in \Gamma$
($\mathcal{S}_@$) $s = @yt$ and $ty \in \Gamma$
($\mathcal{S}_D$) $s = Dt$ and there is some $y \in X$ such that $y \neq x$ and $ty \in \Gamma$
($\mathcal{S}_{\bar D}$) $s = \bar Dt$ and, for every $y \in \mathcal{N}\Gamma$, either $y = x$ or $ty \in \Gamma$

Note that all of the saturatedness conditions but $\mathcal{S}_\Diamond$, $\mathcal{S}_E$, $\mathcal{S}_N$, $\mathcal{S}_D$ and $\mathcal{S}_{\bar D}$ are identical to the corresponding expandedness conditions. $\Gamma$ is called *saturated on a set $X \subseteq \mathcal{N}\Gamma$* if it is saturated on $X$ for all normal formulas $sx \in \Gamma$. Saturated sets are often also called Hintikka sets after the inventor of the concept.

We define $X_{\Gamma,\varphi} := \{x \in \mathcal{N}(\varphi\Gamma) \mid \exists y \in \mathcal{N}\Gamma :\ y \sim_\Gamma x \text{ and } y \text{ active on } \Gamma\}$. The following proposition captures an essential intuition about $X_{\Gamma,\varphi}$.

**Proposition 8.** *Let $\varphi$ be a normalizer for a branch $\Gamma$. If $x$ is active on $\Gamma$, then $\varphi x \in X_{\Gamma,\varphi}$.*

**Proposition 9.** *Let $\varphi$ be a normalizer for a branch $\Gamma$. If $\Gamma$ is open and maximal, then $\varphi\Gamma$ is open and saturated on $X_{\Gamma,\varphi}$.*

*Proof.* First, we show by contradiction that $\varphi\Gamma$ is open. Assume $\varphi\Gamma$ closed. Then there are some $x, y$ such that $\varphi x = \varphi y$ (which is equivalent to $x \sim_\Gamma y$ since $\varphi$ is a normalizer) and either $x \dot{\neq} y \in \Gamma$ or $px, \dot\neg py \in \Gamma$. In the former case, it immediately follows that $\Gamma$ is closed, in contradiction to the assumption. In the latter case, the contradiction follows by Lemma 1.

Now to saturatedness on $X_{\Gamma,\varphi}$. Let us first show that $\mathcal{N}(\mathrm{Mod}\,(\varphi\Gamma)) = \varphi(\mathcal{N}(\mathrm{Mod}\,\Gamma)) \subseteq X_{\Gamma,\varphi}$. Let $x \in \mathcal{N}(\mathrm{Mod}\,\Gamma)$. It suffices to show that $\varphi x \in X_{\Gamma,\varphi}$. By the Subterm Property, $x \in \mathcal{N}(\mathrm{Mod}\,\Gamma_0)$, where $\Gamma_0$ is the initial branch. Since $\Gamma_0$ contains no edges, $x$ is a root of $G_{\Gamma_0}$. Then, by Proposition 4, $x$ is a root of $G_\Gamma$ and hence active on $\Gamma$. Since $x \sim_\Gamma \varphi x$, we have $\varphi x \in X_{\Gamma,\varphi}$.

It remains to show that $\varphi\Gamma$ satisfies the respective saturatedness conditions for all normal formulas $sx \in \Gamma$, which we do by case analysis on $s$. The claim is almost immediate for all cases but $s = \langle\rho\rangle t$, $s = Et$, $s = \dot y$, $s = Dt$ and $s = \bar Dt$, so let us focus on these cases.

**Case** $s = \dot{y}$ **($\mathcal{S}_N$).** It suffices to show that $\varphi y = \varphi x$. By $\mathcal{E}_N$, $y \dot{=} x \in \Gamma$. So, $y \sim_\Gamma x$ and hence $\varphi y = \varphi x$.

**Case** $s = \bar{D}t$ **($\mathcal{S}_{\bar{D}}$).** Similarly to the preceding case.

**Case** $s = \langle\rho\rangle t$ **($\mathcal{S}_\Diamond$).** Let $\varphi x \in X_{\Gamma,\varphi}$. It suffices to show that there is some $y$ such that $(\varphi t)y \in \varphi\Gamma$, $\rho(\varphi x)y \in \Gamma$ or $\tilde{\rho}y(\varphi x) \in \Gamma$, and $\mathcal{L}_{\varphi\Gamma}y = \mathcal{L}_{\varphi\Gamma}z$ for some $z \in X_{\Gamma,\varphi}$.

We know that there is some active $u$ such that $x \sim_\Gamma \varphi x \sim_\Gamma u$. By Lemma 1, $\langle\rho\rangle tu \in \Gamma$. Hence, by $\mathcal{E}_\Diamond$, there is some $v$ such that $tv \in \Gamma$ and either $\rho uv \in \Gamma$ or $\tilde{\rho}vu \in \Gamma$. Since $\varphi u = \varphi x$, $(\varphi t)(\varphi v) \in \varphi\Gamma$ and either $\rho(\varphi x)(\varphi v) \in \varphi\Gamma$ or $\tilde{\rho}(\varphi v)(\varphi x) \in \varphi\Gamma$. So, let $y = \varphi v$. It remains to show that $\mathcal{L}_{\varphi\Gamma}y = \mathcal{L}_{\varphi\Gamma}z$ for some $z \in X_{\Gamma,\varphi}$. Since $u$ is active, by Proposition 6, $v$ is relevant. Hence, by Proposition 7, there is some active $w$ such that $\mathcal{L}_\Gamma v = \mathcal{L}_\Gamma w$. Then, by Lemma 2, $\mathcal{L}_{\varphi\Gamma}y = \mathcal{L}_{\varphi\Gamma}(\varphi w)$. Moreover, by Proposition 8, $\varphi w \in X_{\Gamma,\varphi}$. So, $\varphi w$ is the required $z$.

**Case** $s = Dt$ **($\mathcal{S}_D$).** It suffices to show that there is some $y \in X_{\Gamma,\varphi}$ such that $y \neq \varphi x$ and $(\varphi t)y \in \varphi\Gamma$.

By $\mathcal{E}_D$, there is some $z \in \mathrm{Root}\,\Gamma$ such that $z \not\sim_\Gamma x$ and $tz \in \Gamma$. Then $(\varphi t)(\varphi z) \in \varphi\Gamma$. As $z$ clearly is active on $\Gamma$, by Proposition 8, $\varphi z \in X_{\Gamma,\varphi}$. Moreover, $\varphi z \sim_\Gamma z \not\sim_\Gamma x \sim_\Gamma \varphi x$, i.e. $\varphi z \neq \varphi x$. So, $\varphi z$ is the required $y$.

**Case** $s = Et$ **($\mathcal{S}_E$).** Analogously to the preceding case, but simpler. □

Given a set $\Gamma$ saturated on $X \subseteq \mathcal{N}\Gamma$, we call $\rho xy$ *safe (for $\Gamma$ and $X$)* if $x, y \in X$ and there is some $z \in \mathcal{N}\Gamma$ such that $\rho xz \in \Gamma$ or $\tilde{\rho}zx \in \Gamma$, and $\mathcal{L}_\Gamma z = \mathcal{L}_\Gamma y$. Clearly, if $x, y \in X$ and either $\rho xy \in \Gamma$ or $\tilde{\rho}yx \in \Gamma$, then $\rho xy$ is safe. Moreover, $\rho xy$ is safe if and only if $\tilde{\rho}yx$ is safe.

Let $\Gamma$ be saturated on $X \subseteq \mathcal{N}\Gamma$, and let $x_0 \in X$. We define the modal interpretation $\mathfrak{M}^\Gamma$ as follows:

$$\begin{aligned}
\mathfrak{M}^\Gamma \mathrm{S} &= X\\
\mathfrak{M}^\Gamma x &= \text{if } x \in X \text{ then } x \text{ else } x_0\\
\mathfrak{M}^\Gamma p &= \lambda x \in X.\, \text{if } px \in \Gamma \text{ then } 1 \text{ else } 0\\
\mathfrak{M}^\Gamma r &= \{(x,y) \mid rxy \text{ safe for } \Gamma \text{ and } X\}
\end{aligned}$$

**Proposition 10 (Model Existence).** *Let $\Gamma$ be open and saturated on some $X \subseteq \mathcal{N}\Gamma$. If $x \in X$, $s$ modal and $sx \in \Gamma$, then $\mathfrak{M}^\Gamma$ satisfies $sx$.*

*Proof.* By induction on the size of $s$. The cases $s = p$, $s = \dot{\neg}p$, $s = \dot{y}$, $s = \dot{\neg}\dot{y}$, $s = t_1 \dot{\wedge} t_2$, $s = t_1 \dot{\vee} t_2$, $s = Et$, $s = At$, $s = Dt$ and $s = \bar{D}t$ are easy, so let us focus on the remaining ones.

**Case** $s = \langle\rho\rangle t$**.** By $\mathcal{S}_\Diamond$, there is some $y \in \mathcal{N}\Gamma$ such that $ty \in \Gamma$ and either $\rho xy \in \Gamma$ or $\tilde{\rho}yx \in \Gamma$, and some $z \in X$ such that $\mathcal{L}_\Gamma y = \mathcal{L}_\Gamma z$. So, $\rho xz$ is safe, i.e. $(x,z) \in \mathfrak{M}^\Gamma\rho$. Moreover, $tz \in \Gamma$. Hence, by the inductive hypothesis, $\mathfrak{M}^\Gamma$ satisfies $tz$.

**Case** $s = [\rho]tx$**.** Let $(x,y) \in \mathfrak{M}^\Gamma\rho$. We have to show that $\mathfrak{M}^\Gamma$ satisfies $ty$.

Clearly, $\rho xy$ is safe, so $x, y \in X$ and there is some $z \in \mathcal{N}\Gamma$ such that $\rho xz \in \Gamma$ or $\tilde{\rho}zx \in \Gamma$, and $\mathcal{L}_\Gamma z = \mathcal{L}_\Gamma y$. By $\mathcal{S}_\Box$, it holds $tz \in \Gamma$. Hence $ty \in \Gamma$. The claim follows by the inductive hypothesis. □

Let $\varphi$ be a substitution and $\mathfrak{M}$ a modal interpretation. We define $\mathfrak{M}_\varphi$ to be the modal interpretation obtained from $\mathfrak{M}$ such that, for all terms $s$, $\mathfrak{M}_\varphi s = \mathfrak{M}(\varphi s)$.

**Proposition 11.** *$\mathfrak{M}$ satisfies $\varphi s$ if and only if $\mathfrak{M}_\varphi$ satisfies $s$.*

**Theorem 1 (Model Existence).** *Let $\Gamma$ be open and maximal. Let $\varphi$ be a normalizer for $\Gamma$. If $x$ is active on $\Gamma$ and $sx \in \Gamma$, then $(\mathfrak{M}^{\varphi\Gamma})_\varphi$ satisfies $sx$.*

*Proof.* Let $\Gamma$ be open and maximal. Let $\varphi$ be a normalizer for $\Gamma$. Let $x$ be active on $\Gamma$ and $sx \in \Gamma$. By Proposition 9, $\varphi\Gamma$ is open and saturated on $X_{\Gamma,\varphi}$. By Proposition 8, $\varphi x \in X_{\Gamma,\varphi}$. Then, by Proposition 10, $\mathfrak{M}^{\varphi\Gamma}$ satisfies $\varphi(sx) = (\varphi s)(\varphi x) \in \varphi\Gamma$. Hence, by Proposition 11, $(\mathfrak{M}^{\varphi\Gamma})_\varphi$ satisfies $sx$. ☐

Since all nominals on the initial branch $\Gamma_0$ are roots of $G_{\Gamma_0}$ and hence active, the interpretation constructed in Theorem 1 from any open and maximal extension of $\Gamma_0$ satisfies $\Gamma_0$.

## 7 Explicit Computation of Equational Equivalence

The rule $\mathcal{R}_{\doteq}$, the expandedness conditions $\mathcal{E}_D$ and $\mathcal{E}_{\bar{D}}$, and the closedness criteria for tableau branches take for granted that the equational equivalence relation $\sim_\Gamma$ can be effectively computed. We leave open how this computation is performed. Alternatively, one could make the computation of $\sim_\Gamma$ explicit by replacing $\mathcal{R}_{\doteq}$ by the following two rules.

$$\mathcal{R}_{\doteq}^{\text{sub}}\ \frac{sx \qquad x\doteq y}{sy}\ s \text{ modal} \qquad\qquad \mathcal{R}_{\doteq}^{\text{sym}}\ \frac{x\doteq y}{y\doteq x}$$

Additionally, one could change the closedness criteria and the expandedness conditions for the difference modality to work with an explicit syntactic representation of $\sim_\Gamma$. Note, however, that for the so modified calculus to terminate, one needs to ensure that the computation of $\sim_\Gamma$ is performed before the rule $\mathcal{R}_D$ is applied. One way of doing so is as follows. One takes, in addition to $\mathcal{R}_{\doteq}^{\text{sub}}$ and $\mathcal{R}_{\doteq}^{\text{sym}}$, the following rule.

$$\mathcal{R}_{\doteq}^{\not\doteq}\ \frac{x\not\doteq y \qquad y\doteq z}{x\not\doteq z}$$

One then prioritizes $\mathcal{R}_{\doteq}^{\text{sub}}$, $\mathcal{R}_{\doteq}^{\text{sym}}$ and $\mathcal{R}_{\doteq}^{\not\doteq}$ over $\mathcal{R}_D$ while replacing the conditions "$y \not\sim_\Gamma x$" in $\mathcal{E}_D$ and "$y \sim_\Gamma x$" in $\mathcal{E}_{\bar{D}}$ by "$y\doteq x \notin \Gamma$" and "$y\doteq x \in \Gamma$", respectively, and changing the closedness criterion for disequations from "$x\not\doteq y(\in \Gamma)$ where $x \sim_\Gamma y$" to "$x\not\doteq x \in \Gamma$".

We chose $\mathcal{R}_{\doteq}$ over syntactic rules like $\mathcal{R}_{\doteq}^{\text{sub}}$, $\mathcal{R}_{\doteq}^{\text{sym}}$ and $\mathcal{R}_{\doteq}^{\not\doteq}$ to simplify the presentation and because we didn't want to commit to any particular algorithmic treatment of equational equivalence. Moreover, a practical implementation is likely to use a different, more efficient way of computing $\sim_\Gamma$ than the one suggested by the above rules.

## 8 Conclusion

We have seen a terminating tableau calculus for basic hybrid logic with converse and difference. Termination of the calculus was obtained by combining chain-based blocking for logics with converse as introduced by Horrocks and Sattler [15] with a complete and terminating treatment of $D$ in [10]. To prove completeness of the calculus, it was necessary to refine conventional filtration arguments as found in [15, 14] by distinguishing between modal and equational equivalence of states.

Following [29, 15], one can further extend our calculus to cover reflexive, symmetric and transitive modalities while retaining termination. Since the depth of $G_\Gamma$ is bounded by an exponential in the size of the input, the size of our tableau branches is at most doubly exponential. Hence, a naive implementation would have triply exponential worst-case complexity. Donini and Massacci [30] and later Goré and Nguyen [31] show that caching of satisfiability results for explored tableau branches can reduce the complexity of tableau algorithms for expressive nominal-free description logics to ExpTime, resulting in decision procedures that are worst-case optimal [32, 8]. It is an open problem to find corresponding techniques that would scale to logics with nominals and difference.

**Acknowledgment.** We would like to thank the anonymous referees for their valuable comments and suggestions to this paper.

## References

1. Balbiani, Ph., Demri, S.: Prefixed tableaux systems for modal logics with enriched languages. In Ralescu, A.L., Shanahan, J.G., eds.: Proc. 15th Intl. Joint Conf. on Artificial Intelligence (IJCAI'97), Morgan Kaufmann (1997) 190–195
2. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press (2001)
3. Gargov, G., Goranko, V.: Modal logic with names. Journal of Philosophical Logic **22** (1993) 607–636
4. Areces, C., ten Cate, B.: Hybrid logics. [33]
5. Goranko, V.: Modal definability in enriched languages. Notre Dame Journal of Formal Logic **31**(1) (1990) 81–105
6. de Rijke, M.: The modal logic of inequality. J. Symb. Log. **57**(2) (June 1992) 566–584
7. Venema, Y.: Derivation rules as anti-axioms in modal logic. J. Symb. Log. **58**(3) (1993) 1003–1034
8. Areces, C., Blackburn, P., Marx, M.: The computational complexity of hybrid temporal logics. L. J. of the IGPL **8**(5) (2000) 653–679
9. Fitting, M.: Modal proof theory. [33]
10. Kaminski, M., Smolka, G.: Hybrid tableaux for the difference modality. In: 5th Workshop on Methods for Modalities (M4M-5). (2007)
11. Hughes, G.E., Cresswell, M.J.: An Introduction to Modal Logic. Methuen (1968)
12. Halpern, J.Y., Moses, Y.: A guide to completeness and complexity for modal logics of knowledge and belief. Artif. Intell. **54** (1992) 319–379

13. Horrocks, I., Hustadt, U., Sattler, U., Schmidt, R.: Computational modal logic. [33]
14. Bolander, T., Blackburn, P.: Termination for hybrid tableaus. J. Log. Comput. **17**(3) (2007) 517–554
15. Horrocks, I., Sattler, U.: A description logic with transitive and inverse roles and role hierarchies. J. Log. Comput. **9**(3) (1999) 385–410
16. Horrocks, I., Sattler, U., Tobies, S.: Practical reasoning for very expressive description logics. L. J. of the IGPL **8**(3) (2000) 239–263
17. Horrocks, I., Sattler, U.: A tableau decision procedure for $\mathcal{SHOIQ}$. J. Autom. Reasoning **39**(3) (2007) 249–276
18. Hollunder, B., Baader, F.: Qualifying number restrictions in concept languages. In Allen, J., Fikes, R., Sandewall, E., eds.: Proc. 2nd Intl. Conf. on Principles of Knowledge Representation and Reasoning (KR'91), Morgan Kaufmann (1991) 335–346
19. Calvanese, D., De Giacomo, G., Rosati, R.: A note on encoding inverse roles and functional restrictions in $\mathcal{ALC}$ knowledge bases. In Franconi, E., De Giacomo, G., MacGregor, R.M., Nutt, W., Welty, C.A., eds.: Proc. 1998 Intl. Workshop on Description Logics (DL'98). Volume 11 of CEUR Workshop Proceedings. (1998) 69–71
20. Grädel, E.: On the restraining power of guards. J. Symb. Log. **64**(4) (1999) 1719–1742
21. Ganzinger, H., de Nivelle, H.: A superposition decision procedure for the guarded fragment with equality. In: Proc. 14th Annual IEEE Symposium on Logic in Computer Science (LICS'99), IEEE Computer Society Press (1999) 295–304
22. Gallin, D.: Intensional and Higher-Order Modal Logic. With Applications to Montague Semantics. Volume 19 of Mathematics Studies. North-Holland (1975)
23. Gamut, L.T.F.: Logic, Language and Meaning, Volume 2: Intensional Logic and Logical Grammar. The University of Chicago Press (1991)
24. Carpenter, B.: Type-Logical Semantics. Language, Speech, and Communication. The MIT Press (1997)
25. Hardt, M., Smolka, G.: Higher-order syntax and saturation algorithms for hybrid logic. Electr. Notes Theor. Comput. Sci. **174**(6) (2007) 15–27
26. Benzmüller, C.E., Paulson, L.C.: Exploring properties of normal multimodal logics in simple type theory with LEO-II. In Benzmüller, C.E., Brown, C.E., Siekmann, J., Statman, R., eds.: Festschrift in Honor of Peter B. Andrews on His 70th Birthday. Studies in Logic and the Foundations of Mathematics. IFCoLog To appear.
27. Kaminski, M., Smolka, G.: A straightforward saturation-based decision procedure for hybrid logic. In: Intl. Workshop on Hybrid Logic 2007 (HyLo 2007). (2007)
28. Bolander, T., Braüner, T.: Tableau-based decision procedures for hybrid logic. J. Log. Comput. **16**(6) (2006) 737–763
29. Massacci, F.: Strongly analytic tableaux for normal modal logics. In Bundy, A., ed.: CADE-12. Volume 814 of LNAI., Springer (1994) 723–737
30. Donini, F.M., Massacci, F.: Exptime tableaux for $\mathcal{ALC}$. Artif. Intell. **124**(1) (2000) 87–138
31. Goré, R., Nguyen, L.A.: EXPTIME tableaux with global caching for description logics with transitive roles, inverse roles and role hierarchies. In Olivetti, N., ed.: TABLEAUX'07. Volume 4548 of LNAI., Springer (2007) 133–148
32. Spaan, E.: Complexity of Modal Logics. PhD thesis, ILLC, University of Amsterdam (1993)
33. Blackburn, P., van Benthem, J., Wolter, F., eds.: Handbook of Modal Logic. Volume 3 of Studies in Logic and Practical Reasoning. Elsevier (2006)