Terminating Tableaux for Hybrid Logic with Eventualities

Mark Kaminski and Gert Smolka
Saarland University

June 25, 2010

We present the first terminating tableau system for hybrid logic with eventualities. The system is designed as a basis for gracefully degrading reasoners. Eventualities are formulas of the form $\Diamond^* s$ that hold for a state if it can reach in $n \geq 0$ steps a state satisfying the formula s. The system is prefix-free and employs a novel clausal form that abstracts away from propositional reasoning. It comes with an elegant correctness proof. We discuss some optimizations for decision procedures.

1 Introduction

We consider basic modal logic extended with nominals and eventualities. We call this logic H^*. Nominals are formulas of the form x that hold exactly for the state x. Eventualities are formulas of the form $\Diamond^* s$ that hold for a state if it can reach in $n \geq 0$ steps a state satisfying the formula s. Nominals equip modal logic with equality and are the characteristic feature of hybrid logic [4, 2]. Eventualities extend modal logic with reflexive transitive closure and are an essential feature of PDL [11, 15] and temporal logics [19, 9, 10]. One can see H^* either as hybrid logic extended with eventualities or as stripped-down PDL extended with nominals. Due to the inductive nature of eventualities, H^* is not compact (consider $\Diamond^* \neg p$, p, $\Box p$, $\Box\Box p$, $\ldots$). On the other hand, the satisfiability problem for H^* is decidable and EXPTIME-complete. Decidability can easily be shown with filtration [4]. Decidability in deterministic exponential time follows from a corresponding result for the hybrid μ-calculus [21], a logic that subsumes H^*. EXPTIME-hardness follows from Fischer and Ladner's [11] proof for PDL, which also applies to modal logic with eventualities. See Blackburn et al. [4] for a discussion and an elegant proof (Theorem 6.52).

We are interested in a terminating tableau system for H^* that can serve as a basis for gracefully degrading reasoners. Given that there are terminating tableau systems for both PDL [20, 3, 8, 1, 13] and hybrid logic [7, 5, 16, 17, 18], one would hope that coming up with a terminating system for H^* is not difficult. Once we attacked the problem we found it rather difficult. First of all, the approaches taken by the two families of systems are very different. Hybrid systems rely on fine-grained prefix-based propagation of equational information and lack the structure needed for checking eventualities. PDL systems do not provide the propagation needed for nominals and this propagation is in conflict with the and-or graph representation [20, 13] and the existing techniques for proving correctness.

After some trial an error, we decided to first construct a terminating tableau system for modal logic with eventualities and nothing else. The goal was to obtain a simple system with a simple correctness proof that would scale to the extension with nominals. In the end we found a convincing solution that uses some new ideas. In contrast to existing systems for hybrid logic, our system is prefix-free and does not rely on fine-grained propagation of equational constraints. Following the PDL systems of Baader [3] and De Giacomo and Massacci [8], our system avoids a posteriori eventuality checking by disallowing bad loops. The novel feature of our system is the use of a clausal form that abstracts away from propositional reasoning and puts the focus on modal reasoning. This way termination and bad loop checking become obvious. The crucial part of the correctness proof, which shows that branches with bad loops can be safely ignored, employs the notion of a straight model. A straight model requires that the links on the branch make maximal progress towards the fulfillment of the eventuality they serve. The notion of a straight model evolved in work with Sigurd Schneider [22] and builds on an idea in Baader's [3] correctness proof (Proposition 4.7).

Due to the clausal form, the extension of our system to nominals is straightforward. When we add a new clause to a branch, we add to the new clause all literals that occur in clauses of the branch that have a nominal in common with the new clause. This takes care of nominal propagation. Clauses and links that are already on the branch remain unchanged. Our approach yields a novel and particularly simple tableau system for hybrid logic.

The paper is organized as follows. First, we introduce formulas, interpretations, and the clausal form. We then present the tableau system and its correctness proof in three steps, first for modal logic, then for hybrid logic, and finally for hybrid logic with eventualities. Finally, we discuss some optimizations for decision procedures.

2 The Logic

We assume that two kind of names, called **nominals** and **predicates**, are given. Nominals (x, y) denote states and predicates (p, q) denote sets of states. **Formulas** are defined as follows:

$$s ::= p \mid \neg p \mid s \wedge s \mid s \vee s \mid \Diamond s \mid \Box s \mid \Diamond^* s \mid \Box^* s \mid x \mid \neg x \mid @_x s$$

For simplicity we employ only a single transition relation and consider only formulas in negation normal form. Generalization of our results to multiple transition relations is straightforward. We use the notations $\Diamond^+ s := \Diamond\Diamond^* s$ and $\Box^+ s := \Box\Box^* s$. An **eventuality** is a formula of the form $\Diamond^* s$ or $\Diamond^+ s$. All other **diamond formulas** $\Diamond s$ are called **simple**. An **interpretation** $\mathcal{I}$ consists of the following components:

- A set $|\mathcal{I}|$ of **states**.
- A **transition relation** $\to_{\mathcal{I}} \subseteq |\mathcal{I}| \times |\mathcal{I}|$.
- A set $\mathcal{I}p \subseteq |\mathcal{I}|$ for every predicate p.
- A state $\mathcal{I}x \in |\mathcal{I}|$ for every nominal x.

We write $\to_{\mathcal{I}}^*$ for the reflexive transitive closure of $\to_{\mathcal{I}}$. The **satisfaction relation** $\mathcal{I}, a \vDash s$ between interpretations $\mathcal{I}$, states $a \in |\mathcal{I}|$, and formulas s is defined by induction on s:

$$\begin{array}{rcl@{\qquad}rcl}
\mathcal{I},a \vDash p & \iff & a \in \mathcal{I}p & \mathcal{I},a \vDash s \wedge t & \iff & \mathcal{I},a \vDash s \text{ and } \mathcal{I},a \vDash t \\
\mathcal{I},a \vDash \neg p & \iff & a \notin \mathcal{I}p & \mathcal{I},a \vDash s \vee t & \iff & \mathcal{I},a \vDash s \text{ or } \mathcal{I},a \vDash t \\
\mathcal{I},a \vDash x & \iff & a = \mathcal{I}x & \mathcal{I},a \vDash \Diamond s & \iff & \exists b:\ a \to_{\mathcal{I}} b \text{ and } \mathcal{I},b \vDash s \\
\mathcal{I},a \vDash \neg x & \iff & a \neq \mathcal{I}x & \mathcal{I},a \vDash \Box s & \iff & \forall b:\ a \to_{\mathcal{I}} b \text{ implies } \mathcal{I},b \vDash s \\
\mathcal{I},a \vDash @_x s & \iff & \mathcal{I},\mathcal{I}x \vDash s & \mathcal{I},a \vDash \Box^* s & \iff & \forall b:\ a \to_{\mathcal{I}}^* b \text{ implies } \mathcal{I},b \vDash s \\
 & & & \mathcal{I},a \vDash \Diamond^* s & \iff & \exists b:\ a \to_{\mathcal{I}}^* b \text{ and } \mathcal{I},b \vDash s
\end{array}$$

We interpret sets of formulas conjunctively. Given a set A of formulas, we write $\mathcal{I}, a \vDash A$ if $\mathcal{I}, a \vDash s$ for all formulas $s \in A$. An interpretation $\mathcal{I}$ **satisfies** (or is a **model** of) a formula s or a set A of formulas if there is a state $a \in |\mathcal{I}|$ such that $\mathcal{I}, a \vDash s$ or, respectively, $\mathcal{I}, a \vDash A$. A formula s (a set A) is **satisfiable** if s (A) has a model.

We say that two formulas s and t are **equivalent** and write $s \cong t$ if the equivalence $\mathcal{I}, a \vDash s \iff \mathcal{I}, a \vDash t$ holds for all interpretations $\mathcal{I}$ and all states $a \in |\mathcal{I}|$. Two important equivalences are $\Diamond^* s \cong s \vee \Diamond^+ s$ and $\Box^* s \cong s \wedge \Box^+ s$.

We write $\mathrm{H}^*_@$ for the full logic and define several sublogics:

K	$p \mid \neg p \mid s \wedge s \mid s \vee s \mid \Diamond s \mid \Box s \mid \Box^* s$
K^*	K extended with $\Diamond^* s$
H	K extended with x, $\neg x$
H^*	H extended with $\Diamond^* s$
$\mathrm{H}^*_@$	H^* extended with $@_x s$

Note that K is basic modal logic plus positive occurrences of $\Box^* s$, and H is basic hybrid logic plus positive occurrences of $\Box^* s$.

3 Clausal Form

We define a clausal form for our logic. The clausal form allows us to abstract from propositional reasoning and to focus on modal reasoning.

A **literal** is a formula of the form p, $\neg p$, $\Diamond s$, $\Box s$, x, $\neg x$, or $@_x s$. A **clause** (C, D) is a finite set of literals that contains no complementary pair (p and $\neg p$ or x and $\neg x$). Clauses are interpreted conjunctively. **Satisfaction of clauses** (i.e., $\mathcal{I}, a \models C$) is a special case of satisfaction of sets of formulas (i.e., $\mathcal{I}, a \models A$), which was defined in §2. For instance, the clause $\{\Diamond p, \Box \neg p\}$ is unsatisfiable. Note that every clause not containing literals of the forms $\Diamond s$ and $@_x s$ is satisfiable. We will show that for every formula one can compute $n \geq 0$ clauses such that the disjunction of the clauses is equivalent to the formula.

The **syntactic closure** SA of a set A of formulas is the least set of formulas that contains A and is closed under the rules

$$\frac{\neg s}{s} \qquad \frac{s \wedge t}{s\,,\,t} \qquad \frac{s \vee t}{s\,,\,t} \qquad \frac{\Diamond s}{s} \qquad \frac{\Box s}{s} \qquad \frac{\Box^* s}{s\,,\,\Box^+ s} \qquad \frac{\Diamond^* s}{s\,,\,\Diamond^+ s} \qquad \frac{@_x s}{x\,,\,s}$$

Note that SA is finite if A is finite, and that the size of SA is linear in the size of A (sum of the sizes of the formulas appearing as elements of A).

The **support relation** $C \rhd s$ between clauses C and formulas s is defined by induction on s:

$$\begin{aligned}
C \rhd s &\iff s \in C \quad \text{if } s \text{ is a literal} \\
C \rhd s \wedge t &\iff C \rhd s \text{ and } C \rhd t \\
C \rhd s \vee t &\iff C \rhd s \text{ or } C \rhd t \\
C \rhd \Box^* s &\iff C \rhd s \text{ and } C \rhd \Box^+ s \\
C \rhd \Diamond^* s &\iff C \rhd s \text{ or } C \rhd \Diamond^+ s
\end{aligned}$$

We say C **supports** s if $C \rhd s$. We write $C \rhd A$ and say C **supports** A if $C \rhd s$ for every $s \in A$. Note that $C \rhd D \iff D \subseteq C$ if C and D are clauses.

Proposition 3.1 If $\mathcal{I}, a \vDash C$ and $C \rhd A$, then $\mathcal{I}, a \vDash A$.

Proposition 3.2 If $C \rhd A$ and $C \subseteq D$ and $B \subseteq A$, then $D \rhd B$.

We define a function $\mathcal{D}$ that yields for every set A of formulas the set of all minimal clauses supporting A:

$$\mathcal{D}A \; := \; \{\, C \mid C \rhd A \text{ and } \forall D \subseteq C\colon\; D \rhd A \text{ implies } D = C \,\}$$

We call $\mathcal{D}A$ the **DNF** of A.

Example 3.3 Consider $s = p \wedge q \vee \neg p \wedge q$. Then $\mathcal{D}\{s\} = \{\{p,q\},\{\neg p,q\}\}$. Hence $\{q\} \not\rhd \{s\}$ even though q and s are equivalent. □

If X is a set, we use the notation $X\,;x := X \cup \{x\}$.

Proposition 3.4

1. $\mathcal{I}, a \vDash A \iff \exists C \in \mathcal{D}A\colon\; \mathcal{I}, a \vDash C$.
2. If $C \in \mathcal{D}A$, then $C \subseteq \mathcal{S}A$.
3. $C \rhd A \iff \exists D \in \mathcal{D}A\colon\; D \subseteq C$.
4. $\mathcal{D}(A\,;s) \subseteq \mathcal{D}(A\,;\Diamond^* s)$.

Proposition 3.5 If A is a finite set of formulas, then $\mathcal{D}A$ is finite.

Proof The claim follows with Proposition 3.4 (2) since $\mathcal{S}A$ is finite.

The DNF of a finite set of formulas can be computed with the following tableau rules:

$$\frac{s \wedge t}{s\,,\,t} \qquad \frac{s \vee t}{s \mid t} \qquad \frac{\Box^* s}{s\,,\,\Box^+ s} \qquad \frac{\Diamond^* s}{s \mid \Diamond^+ s}$$

To obtain $\mathcal{D}A$, one develops A into a complete tableau. The literals of each open branch yield a clause. The minimal clauses obtained this way constitute $\mathcal{D}A$.

Let C and D be clauses. The **request of C** is $\mathcal{R}C := \{\, t \mid \Box t \in C \,\}$. We say **$D$ realizes $\Diamond s$ in C** if $D \rhd \mathcal{R}C\,;s$.

Proposition 3.6 If $\Diamond s \in C$ and $\mathcal{I}$ satisfies C, then $\mathcal{I}$ satisfies some clause $D \in \mathcal{D}(\mathcal{R}C\,;s)$.

Proof Follows with Proposition 3.4 (1).

4 Tableaux for K

We start with a terminating tableau system for the sublogic K to demonstrate the basic ideas of our approach. A **branch** of the system is a finite and nonempty set of clauses. A **model of a branch** is an interpretation that satisfies all clauses of the branch. A branch is **satisfiable** if it has a model. Let Γ be a branch, C be a clause, and $\Diamond s$ be a literal. We say that

- **Γ realizes $\Diamond s$ in C** if $D \rhd \mathcal{R}C\,;s$ for some clause $D \in \Gamma$.
- **Γ is evident** if Γ realizes $\Diamond s$ in C for all $\Diamond s \in C \in \Gamma$.

The **syntactic closure** $\mathcal{S}\Gamma$ of a branch Γ is the union of the syntactic closures of the clauses $C \in \Gamma$. Note that the syntactic closure of a branch is finite. Moreover, $C \subseteq \mathcal{S}\Gamma$ for all clauses $C \in \Gamma$.

Every evident branch describes a finite interpretation that satisfies all its clauses. The states of the interpretation are the clauses of the branch, and the transitions of the interpretation are the pairs (C, D) such that $D \rhd \mathcal{R}C$.

Theorem 4.1 (Model Existence) Every evident branch has a finite model.

Proof Let Γ be an evident branch and $\mathcal{I}$ be an interpretation as follows:

- $|\mathcal{I}| = \Gamma$
- $C \to_{\mathcal{I}} D \iff D \rhd \mathcal{R}C$
- $C \in \mathcal{I}p \iff p \in C$

We show $\forall s \in \mathcal{S}\Gamma\ \forall C \in \Gamma\colon\ C \rhd s \implies \mathcal{I}, C \vDash s$ by induction on s. Let $s \in \mathcal{S}\Gamma$, $C \in \Gamma$, and $C \rhd s$. We show $\mathcal{I}, C \vDash s$ by case analysis. The cases are all straightforward except possibly for $s = \Box^* t$. So let $s = \Box^* t$. Let $C = C_1 \to_{\mathcal{I}} \ldots \to_{\mathcal{I}} C_n$. We show $\mathcal{I}, C_n \vDash t$ by induction on n. If $n = 1$, we have $C_n \rhd s$ by assumption. Hence $C_n \rhd t$ and the claim follows by the outer inductive hypothesis. If $n > 1$, we have $s \in \mathcal{R}C_1$ since $\Box s \in C_1$ since $C_1 \rhd \Box s$ since $C_1 \rhd s$. Thus $C_2 \rhd s$ and the claim follows by the inner inductive hypothesis.

The tableau system for K is obtained with a single rule.

Expansion Rule for K
If $\Diamond s \in C \in \Gamma$ and Γ does not realize $\Diamond s$ in C,
then expand Γ to all branches $\Gamma\,;D$ such that $D \in \mathcal{D}(\mathcal{R}C; s)$.

The expansion rule for K has the obvious property that it applies to a branch if and only if the branch is not evident. It is possible that the expansion rule applies to a branch but does not produce an extended branch. We call a branch **closed** if this is the case. Note that a branch is closed if and only if it contains a clause C that contains a literal $\Diamond s$ such that the DNF of $\mathcal{R}C; s$ is empty.

Example 4.2 Consider the clause $C = \{\Diamond p,\ \Box\neg p\}$. Since $\Diamond p$ is not realized in C in $\Gamma = \{C\}$, the expansion rule applies to the branch Γ. Since $\mathcal{D}(\mathcal{R}C\,;p) = \emptyset$, the expansion rule fails to produce an extension of Γ. Thus Γ is closed. $\square$

Example 4.3 Here is a complete tableau for a clause C_1.

$$\begin{array}{r|l} \multicolumn{2}{c}{C_1 = \{\Diamond\Diamond p,\ \Diamond(q \wedge \Diamond p),\ \Box(q \vee \Box\neg p)\}} \\ \hline C_2 = \{\Diamond p, q\} & C_3 = \{\Diamond p, \Box\neg p\} \\ C_4 = \{p\} & \end{array}$$

The development of the tableau starts with the branch $\{C_1\}$. Application of the expansion rule to $\Diamond\Diamond p \in C_1$ yields the branches $\{C_1, C_2\}$ and $\{C_1, C_3\}$. The branch $\{C_1, C_3\}$ is closed. Expansion of $\Diamond p \in C_2 \in \{C_1, C_2\}$ yields the evident branch $\{C_1, C_2, C_4\}$. Note that C_2 realizes $\Diamond(q \wedge \Diamond p)$ in C_1. $\square$

Theorem 4.4 (Termination) The tableau system for K terminates.

Proof Let a branch Γ' be obtained from a branch Γ by the expansion rule such that $\Gamma \subsetneq \Gamma'$. By Proposition 3.4 (2) we have $\mathcal{S}\Gamma' = \mathcal{S}\Gamma$ and $\Gamma \subsetneq \Gamma' \subseteq 2^{\mathcal{S}\Gamma'} = 2^{\mathcal{S}\Gamma}$. This suffices for termination since $\mathcal{S}\Gamma$ is finite.

Theorem 4.5 (Soundness) Let $\mathcal{I}$ be a model of a branch Γ and $\Diamond s \in C \in \Gamma$. Then there exists a clause $D \in \mathcal{D}(\mathcal{R}C\,;s)$ such that $\Gamma;D$ is a branch and $\mathcal{I}$ is a model of $\Gamma;D$.

Proof Follows with Proposition 3.6.

We now have a tableau-based decision procedure that decides the satisfiability of branches. Given a branch Γ, the procedure either extends Γ to an evident branch that describes a model of Γ, or it constructs a closed tableau that proves that Γ is unsatisfiable. The procedure is recursive. It checks whether the current branch contains a diamond formula that is not yet realized. If no such formula exists, the branch is evident and the procedure terminates. Otherwise, the DNF of the body of such a formula and the request of the clause containing it are computed. If the DNF is empty, the branch is closed and thus unsatisfiable. Otherwise, the branch is expanded into as many branches as the DNF has clauses and the procedure continues recursively. The correctness of the procedure follows from the theorems and propositions stated above.

5 Tableaux for H

We now develop a terminating tableau system for the sublogic H, which extends K with nominals. The system is very different from existing systems for hybrid logic [7, 5, 16, 17, 18] since it does not employ prefixes. The key observation is that two clauses that contain a common nominal must be satisfied by the same state in every model of the branch. Thus if two clauses on a branch contain a common nominal, we can always add the union of the two clauses to the branch.

Proposition 5.1 Suppose an interpretation $\mathcal{I}$ satisfies two clauses C and D that contain a common nominal $x \in C \cap D$. Then $\mathcal{I}$ satisfies $C \cup D$ and the set $C \cup D$ is a clause.

We call a clause **nominal** if it contains a nominal. Let Γ be a set of clauses and A be a set of formulas. We define two notations to realize what we call **nominal propagation**:

$$A^\Gamma := A \cup \{\, s \mid \exists x \in A\ \exists C \in \Gamma\colon\ x \in C \wedge s \in C \,\}$$
$$\mathcal{D}^\Gamma A := \{\, C^\Gamma \mid C \in \mathcal{D}A \text{ and } C^\Gamma \text{ is a clause} \,\}$$

Note that A^Γ is the least set of formulas that contains A and all clauses $C \in \Gamma$ that have a nominal in common with A. Thus $(A^\Gamma)^\Gamma = A^\Gamma$. Moreover, $A^\Gamma = A$ if A contains no nominal.

Proposition 5.2 Let A be a set of formulas, $\mathcal{I}$ be a model of a branch Γ, and a be a state of $\mathcal{I}$. Then $\mathcal{I}, a \models A \iff \mathcal{I}, a \models A^\Gamma$.

Proof Follows with Proposition 5.1.

The **branches** of the tableau system for H are finite and nonempty sets Γ of clauses that satisfy the following condition:

- **Nominal coherence:** If $C \in \Gamma$, then $C^\Gamma \in \Gamma$.

Satisfaction, **realization**, **evidence**, and the **syntactic closure** of branches are defined as for K. The **core** $\mathcal{C}\Gamma := \{\, C \in \Gamma \mid C^\Gamma = C \,\}$ of a branch Γ is the set of all clauses of Γ that are either maximal or not nominal.

Proposition 5.3 Let Γ be a branch. Then:

1. $\mathcal{C}\Gamma$ is a branch.
2. An interpretation satisfies Γ iff it satisfies $\mathcal{C}\Gamma$.
3. Γ is evident iff $\mathcal{C}\Gamma$ is evident.

Proof Claims (1) and (2) are obvious, and (3) follows with Proposition 3.2.

Theorem 5.4 (Model Existence) Every evident branch has a finite model.

Proof Let Γ be an evident branch. Without loss of generality we can assume that for every nominal $x \in \mathcal{S}\Gamma$ there is a unique clause $C \in \mathcal{C}\Gamma$ such that $x \in C$ (add clauses $\{x\}$ as necessary). We choose an interpretation $\mathcal{I}$ that satisfies the conditions

- $|\mathcal{I}| = \mathcal{C}\Gamma$
- $C \rightarrow_{\mathcal{I}} D \iff D \rhd \mathcal{R}C$
- $C \in \mathcal{I}p \iff p \in C$
- $\mathcal{I}x = C \iff x \in C \qquad$ for all $x \in \mathcal{S}\Gamma$

The last condition can be satisfied since Γ is nominally coherent. We show $\forall s \in \mathcal{S}\Gamma\ \forall C \in \mathcal{C}\Gamma\colon\ C \rhd s \implies \mathcal{I}, C \vDash s$ by induction on s. Let $s \in \mathcal{S}\Gamma$, $C \in \mathcal{C}\Gamma$, and $C \rhd s$. We show $\mathcal{I}, C \vDash s$ by case analysis. The proof now proceeds as the proof of Theorem 4.1. The additional cases for nominals can be argued as follows.

Let $s = x$. Then $x \in C$ and thus $\mathcal{I}x = C$. Hence $\mathcal{I}, C \vDash s$.

Let $s = \neg x$. Then $\neg x \in C$ and thus $x \notin C$ and $\mathcal{I}x \neq C$. Hence $\mathcal{I}, C \vDash s$.

Nominal coherence acts as an invariant for the tableau system for H. The expansion rule for H refines the expansion rule for K such that the invariant is maintained. The tableau system for H is obtained with the following rule.

Expansion Rule for H
If $\Diamond s \in C \in \mathcal{C}\Gamma$ and Γ does not realize $\Diamond s$ in C,
then expand Γ to all branches $\Gamma; D$ such that $D \in \mathcal{D}^{\Gamma}(\mathcal{R}C; s)$.

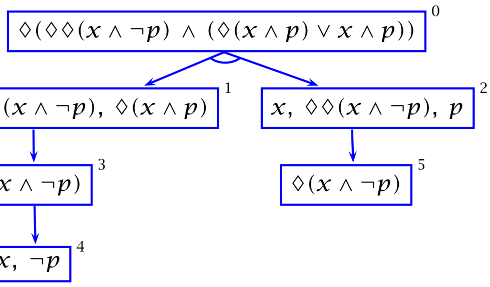
As in the system for K, the expansion rule for H has the property that it applies to a branch if and only if the branch is not evident. Moreover, termination follows as for K. The adaption of the soundness theorem is also straightforward.

Theorem 5.5 (Soundness) Let $\mathcal{I}$ be a model of a branch Γ and $\Diamond s \in C \in \Gamma$. Then there exists a clause $D \in \mathcal{D}^{\Gamma}(\mathcal{R}C; s)$ such that $\Gamma; D$ is a branch and $\mathcal{I}$ is a model of $\Gamma; D$.

Proof Since $\mathcal{I}$ satisfies C, we know by Proposition 3.6 that $\mathcal{I}$ satisfies some clause $D \in \mathcal{D}(\mathcal{R}C; s)$. The claim follows with Proposition 5.2.

We have now arrived at a decision procedure for the sublogic H.

Example 5.6 Consider the following closed tableau.



The numbers identifying the clauses indicate the order in which they are introduced. Once clause 4 is introduced, $\Diamond(x \wedge p)$ in clause 1 cannot be realized due to nominal propagation from clause 4. Hence the left branch is closed. The right branch is also closed since the diamond formula in clause 5 cannot be realized due to nominal propagation from clause 2. $\square$

The example shows that nominals have a severe impact on modal reasoning. The impact of nominals is also witnessed by the fact that in K the union of two branches is a branch while this is not the case in H (e.g., $\{\{x, p\}\}$ and $\{\{x, \neg p\}\}$).

Remark 5.7 To obtain the optimal worst-case run time for tableau provers, one must avoid recomputation. In the absence of nominals this can be accomplished with a minimal and-or graph representation [20, 13]. Unfortunately, the minimal and-or graph representation is not compatible with nominal propagation. To see this, consider the minimal and-or graph representing the tableau of Example 5.6. This graph represents clauses 3 and 5 with a single node. This is not correct since the nominal context of the clauses is different. While clause 3 can be expanded, clause 5 cannot. $\square$

6 Evidence for H^*

Next, we consider the sublogic H^*, which extends H with eventualities $\Diamond^* s$. We define branches and evidence for H^* and prove the corresponding model existence theorem. As one would expect, realization of eventualities $\Diamond^+ s$ is more involved than realization of simple diamond formulas.

Example 6.1 Consider the clause $C = \{\Diamond^+ \neg p,\ \Box^+ p,\ p\}$. We have $\mathcal{R}C = \{\Box^* p\}$. Hence $C \rhd \mathcal{R}C; \Diamond^* \neg p$. If we extend our definitions for H to H^*, the branch $\{C\}$ is evident. However, the clause C is not satisfiable. $\square$

To obtain an adequate notion of realization for eventualities, branches for H^* will contain links in addition to clauses. A **link** is a triple $C(\Diamond s)D$ such that

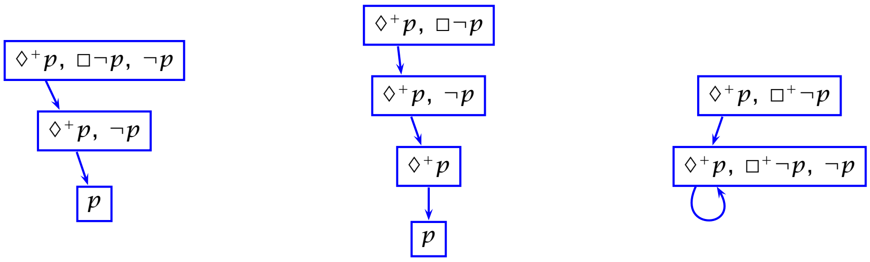


Figure 1: Three quasi-branches drawn as graphs

$\Diamond s \in C$ and $D \rhd \mathcal{R}C; s$. A **quasi-branch** Γ is a finite and nonempty set of clauses and links such that Γ contains the clauses C and D if it contains a link $C(\Diamond s)D$. A **model of a quasi-branch** is an interpretation that satisfies all of its clauses. Note that a model is not required to satisfy the links of a quasi-branch. A quasi-branch is **satisfiable** if it has a model. A quasi-branch Γ **realizes** $\Diamond s$ **in** C if Γ contains some link $C(\Diamond s)D$. Quasi-branches can be drawn as graphs with links pointing from diamond formulas to clauses. Figure 1 shows three examples. The first two graphs describe models of the clauses. This is not the case for the rightmost graph where both clauses are unsatisfiable. Still, all diamond formulas are realized with links. The problem lies in the "bad loop" that leads from the lower clause to itself.

The notations A^Γ and $\mathcal{D}^\Gamma A$ are defined for quasi-branches in the same way as they are defined for the branches of H. Let Γ be a quasi-branch. A **path for** $\Diamond^+ s$ **in** Γ is a sequence $C_1 \dots C_n$ of clauses such that $n \geq 2$ and:

1. $\forall i \in [1, n]\colon\ \ C_i^\Gamma = C_i$.
2. $\forall i \in [1, n-1]\ \exists D\colon\ \ C_i(\Diamond^+ s)D \in \Gamma$ and $D^\Gamma = C_{i+1}$.
3. $\forall i \in [2, n-1]\colon\ \ C_i \not\rhd s$.

A **run for** $\Diamond^+ s$ **in** C **in** Γ is a path $C \dots D$ for $\Diamond^+ s$ in Γ such that $D \rhd s$. A **bad loop for** $\Diamond^+ s$ **in** Γ is a path $C \dots C$ for $\Diamond^+ s$ in Γ such that $C \not\rhd s$. A **branch** is a quasi-branch Γ that satisfies the following conditions:

- **Nominal coherence:** If $C \in \Gamma$, then $C^\Gamma \in \Gamma$.
- **Functionality:** If $C(\Diamond s)D \in \Gamma$ and $C(\Diamond s)E \in \Gamma$, then $D = E$.
- **Bad-loop-freeness:** There is no bad loop in Γ.

The first two quasi-branches in Fig. 1 are branches. The third quasi-branch in Fig. 1 is not a branch since it contains a bad loop.

The **core** $\mathcal{C}\Gamma$ of a branch Γ is $\mathcal{C}\Gamma := \{\, C \in \Gamma \mid C^\Gamma = C \,\}$. A branch Γ is **evident** if Γ realizes $\Diamond s$ in C for all $\Diamond s \in C \in \mathcal{C}\Gamma$. The **syntactic closure** $\mathcal{S}\Gamma$ of a branch Γ is the union of the syntactic closures of the clauses $C \in \Gamma$.

Proposition 6.2 Let Γ be an evident branch and $\Diamond^+ s \in C \in \mathcal{C}\Gamma$. Then there is a unique run for $\Diamond^+ s$ in C in Γ.

Proof Since Γ realizes every eventuality in every clause in $\mathcal{C}\Gamma$ and Γ is finite, functional and bad-loop-free, there is a unique run for $\Diamond^+ s$ in C in Γ.

Theorem 6.3 (Model Existence) Every evident branch has a finite model.

Proof Let Γ be an evident branch. Without loss of generality we can assume that for every nominal $x \in \mathcal{S}\Gamma$ there is a unique clause $C \in \mathcal{C}\Gamma$ such that $x \in C$ (add clauses $\{x\}$ as necessary). We choose an interpretation $\mathcal{I}$ that satisfies the following conditions:

- $|\mathcal{I}| = \mathcal{C}\Gamma$
- $C \to_{\mathcal{I}} D \iff \exists s, E\colon\ C(\Diamond s)E \in \Gamma$ and $D = E^{\Gamma}$
- $C \in \mathcal{I}p \iff p \in C$
- $\mathcal{I}x = C \iff x \in C$ for all $x \in \mathcal{S}\Gamma$

We show $\forall s \in \mathcal{S}\Gamma\ \forall C \in \mathcal{C}\Gamma\colon\ C \rhd s \implies \mathcal{I}, C \vDash s$ by induction on s. Let $s \in \mathcal{S}\Gamma$, $C \in \mathcal{C}\Gamma$, and $C \rhd s$. We show $\mathcal{I}, C \vDash s$ by case analysis. Except for $s = \Diamond^* t$ the claim follows as in the proofs of Theorems 4.1 and 5.4.

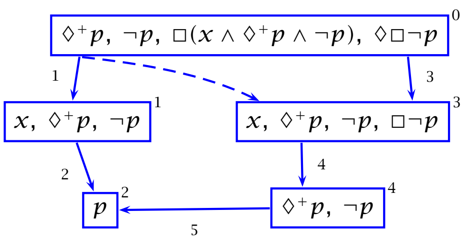
Let $s = \Diamond^* t$. Since $C \rhd s$, we have either $C \rhd t$ or $C \rhd \Diamond^+ t$. If $C \rhd t$, then $\mathcal{I}, C \vDash t$ by the inductive hypothesis and the claim follows. Otherwise, let $C \rhd \Diamond^+ t$. Then $\Diamond^+ t \in C \in \mathcal{C}\Gamma$. By Proposition 6.2 we know that there is a run for $\Diamond^+ t$ in C in Γ. Thus $C \to_{\mathcal{I}}^* D$ and $D \rhd t$ for some clause $D \in \mathcal{C}\Gamma$. Hence $\mathcal{I}, D \vDash t$ by the inductive hypothesis. The claim follows.

7 Tableaux for H*

The tableau system for H* is obtained with the following rule.

Expansion Rule for H^*
If $\Diamond s \in C \in \mathcal{C}\Gamma$ and Γ does not realize $\Diamond s$ in C,
then expand Γ to all branches $\Gamma; D; C(\Diamond s)D$ such that $D \in \mathcal{D}^{\Gamma}(\mathcal{R}C; s)$.

Example 7.1 Here is a tableau derivation of an evident branch from an initial clause with eventualities.

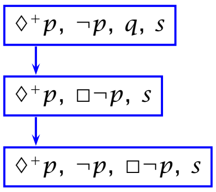


The numbers indicate the order in which the links and clauses are introduced. In the final branch, the clauses 0, 3, 4, 2 constitute a run for $\Diamond^+ p$ in clause 0.

The dashed link is not on the branch. We use it to indicate the implicit redirection of link 1 that occurs when clause 3 is added. The implicit redirection is due to nominal propagation and is realized in the definition of paths. Note that before link 3 is added, the clauses 0, 1, 2 constitute a run for $\Diamond^+ p$ in clause 0. When clause 3 is added, this run disappears since clause 1 is no longer in the core. □

As in the system for H, the expansion rule for H* has the property that it applies to a branch if and only if the branch is not evident. Moreover, termination follows essentially as for H. There is, however, an essential difference as it comes to soundness. Due to our definition of branches a candidate extension $\Gamma; D; C(\Diamond s)D$ is only admissible if it is a bad-loop-free quasi-branch. We now encounter the difficulty that the analogue of the soundness property for H (Theorem 5.5) does not hold since there are satisfiable branches to which the expansion rule applies but fails to produce extended branches since all candidate branches contain bad loops. This is shown by the next example.

Example 7.2 Consider the literal $s := \Box^+(q \vee \Box\neg p)$ and the following branch:



Note that the branch is satisfiable. Since the eventuality in the third clause is not realized, the branch is not evident. The expansion rule applies to the eventuality in the third clause but does not produce an extended branch since both candidate extensions contain bad loops (one extension adds a link from the third clause to the first clause, and the other adds a link from the third clause to itself).

Note that the branch can be obtained by starting with the first clause. The link for the literal $\Diamond^+ p$ in the first clause does not make progress and leads to the bad loop situation. There are two alternative links for this literal that point to the clauses $\{p,\ q,\ s\}$ and $\{p,\ \Box\neg p,\ s\}$. Both yield evident branches. □

We solve the problem with the notion of a *straight link*. The idea is that a straight link for an eventuality $\Diamond^+ s$ reduces the distance to a clause satisfying s as much as possible. We will define straightness with respect to a model.

Let $\mathcal{I}$ be an interpretation, A be a set of formulas, and s be a formula. The **distance from A to s in $\mathcal{I}$** is defined as follows:

$$\delta_{\mathcal{I}} A s \ := \ \min\{\, n \in \mathbb{N} \mid \exists a, b\colon\ a \to^n_{\mathcal{I}} b \text{ and } \mathcal{I}, a \vDash A \text{ and } \mathcal{I}, b \vDash s \,\}$$

where $\min \emptyset = \infty$ and $n < \infty$ for all $n \in \mathbb{N}$. The relations $\to_{\mathcal{I}}^{n}$ are defined as usual: $a \to_{\mathcal{I}}^{0} b$ iff $a = b$ and $a \in |\mathcal{I}|$, and $a \to_{\mathcal{I}}^{n+1} b$ iff $a \to_{\mathcal{I}} a'$ and $a' \to_{\mathcal{I}}^{n} b$ for some a'.

Proposition 7.3 $\delta_{\mathcal{I}} As < \infty$ iff $\mathcal{I}$ satisfies $A; \Diamond^* s$.

Proposition 7.4 Let $\mathcal{I}$ be a model of a quasi-branch Γ. Then $\delta_{\mathcal{I}} As = \delta_{\mathcal{I}} A^{\Gamma} s$.

A link $C(\Diamond^+ s)D$ is **straight** for an interpretation $\mathcal{I}$ if the following conditions are satisfied:

1. $\delta_{\mathcal{I}} Ds \le \delta_{\mathcal{I}} Es$ for every $E \in \mathcal{D}(\mathcal{R}C; \Diamond^* s)$.
2. If $\delta_{\mathcal{I}} Ds = 0$, then $D \rhd s$.

A **straight model** of a quasi-branch Γ is a model $\mathcal{I}$ of Γ such that every link $C(\Diamond^+ s)D \in \Gamma$ is straight for $\mathcal{I}$.

Lemma 7.5 (Straightness) A quasi-branch that has a straight model does not have a bad loop.

Proof By contradiction. Let $\mathcal{I}$ be a straight model of a quasi-branch Γ and let $C_1 \dots C_n$ be a bad loop for $\Diamond^+ s$ in Γ. Then $C_n = C_1$ and $n \ge 2$. To obtain a contradiction, it suffices to show that $\delta_{\mathcal{I}} C_i s > \delta_{\mathcal{I}} C_{i+1} s$ for all $i \in [1, n-1]$. Let $i \in [1, n-1]$.

1. We have $C_i \in \mathcal{C}\Gamma$, $C_i \not\rhd s$, $C_i(\Diamond^+ s)D \in \Gamma$, and $D^{\Gamma} = C_{i+1}$ for some $D \in \Gamma$.
2. We show $\delta_{\mathcal{I}} C_i s < \infty$. By (1) we have $\Diamond^+ s \in C_i \in \Gamma$. The claim follows by Proposition 7.3 since $\mathcal{I}$ satisfies C_i.
3. We show $0 < \delta_{\mathcal{I}} C_i s$. Case analysis.
   a) $i > 1$. Then $C_{i-1}(\Diamond^+ s)E \in \Gamma$ and $E^{\Gamma} = C_i$ for some E. By Proposition 7.4 and the second condition for straight links it suffices to show $E \not\rhd s$. This holds by Proposition 3.2 since $C_i \not\rhd s$ by (1).
   b) $i = 1$. Then $C_{n-1}(\Diamond^+ s)E \in \Gamma$ and $E^{\Gamma} = C_1$ for some E. By Proposition 7.4 and the second condition for straight links it suffices to show $E \not\rhd s$. This holds by Proposition 3.2 since $C_1 \not\rhd s$ by (1).
4. By (2) and (3) there are states a, b, c such that $\mathcal{I}, a \vDash C_i$, $a \to_{\mathcal{I}} b \to_{\mathcal{I}}^{\delta_{\mathcal{I}} C_i s - 1} c$ and $\mathcal{I}, c \vDash s$. We have $\mathcal{I}, b \vDash \mathcal{R}C_i ; \Diamond^* s$. By Proposition 3.4(1) there is a clause $E \in \mathcal{D}(\mathcal{R}C_i ; \Diamond^* s)$ such that $\mathcal{I}, b \vDash E$. Thus $\delta_{\mathcal{I}} Es \le \delta_{\mathcal{I}} C_i s - 1$. By Proposition 7.4 and the first condition for straight links we have $\delta_{\mathcal{I}} C_{i+1} s = \delta_{\mathcal{I}} Ds \le \delta_{\mathcal{I}} Es < \delta_{\mathcal{I}} C_i s$, which yields the claim. ■

Theorem 7.6 (Soundness) Let $\mathcal{I}$ be a straight model of a branch Γ and let $\Diamond s \in C \in \Gamma$. Moreover, let Γ not realize $\Diamond s$ in C. Then there exists a clause $D \in \mathcal{D}^{\Gamma}(\mathcal{R}C ; s)$ such that $\Gamma ; D ; C(\Diamond s)D$ is a branch and $\mathcal{I}$ is a straight model of $\Gamma ; D ; C(\Diamond s)D$.

Proof Since $\mathcal{I}$ is a model of C and $\Diamond s \in C$, there is a clause $D \in \mathcal{D}^\Gamma(\mathcal{R}C;s)$ that is satisfied by $\mathcal{I}$ (Propositions 3.6 and 5.2). For every such clause, $\Gamma;D;C(\Diamond s)D$ is a quasi-branch that has $\mathcal{I}$ as a model and satisfies the nominal coherence and functionality conditions. By Lemma 7.5 it suffices to show that we can choose D such that $\mathcal{I}$ is straight for $C(\Diamond s)D$. If $\Diamond s$ is not an eventuality, this is trivially the case. Otherwise, let $\Diamond s = \Diamond^+ t$ and $s = \Diamond^* t$. We proceed by case analysis.

1. $\mathcal{I}$ satisfies $\mathcal{R}C;t$. We pick a clause $D \in \mathcal{D}^\Gamma(\mathcal{R}C;t)$ that is satisfied by $\mathcal{I}$. By Proposition 3.4 (4), we have $\mathcal{D}(\mathcal{R}C;t) \subseteq \mathcal{D}(\mathcal{R}C;\Diamond^* t)$, and consequently $\mathcal{D}^\Gamma(\mathcal{R}C;t) \subseteq \mathcal{D}^\Gamma(\mathcal{R}C;\Diamond^* t)$. Thus $D \in \mathcal{D}^\Gamma(\mathcal{R}C;\Diamond^* t)$ as required. It remains to show that $\mathcal{I}$ is straight for $C(\Diamond^+ t)D$. This is the case since $\delta_{\mathcal{I}} Dt = 0$ since $D \rhd t$ and $\mathcal{I}$ satisfies D (Proposition 3.1).
2. $\mathcal{I}$ does not satisfy $\mathcal{R}C;t$. This time we choose $D \in \mathcal{D}^\Gamma(\mathcal{R}C;\Diamond^* t)$ such that $\mathcal{I}$ satisfies D and $\delta_{\mathcal{I}} Dt$ is minimal. We show that $\mathcal{I}$ is straight for $C(\Diamond^+ t)D$.

   Let $E \in \mathcal{D}(\mathcal{R}C;\Diamond^* t)$. We show $\delta_{\mathcal{I}} Dt \le \delta_{\mathcal{I}} Et$. If $\mathcal{I}$ does not satisfy E, the claim holds by Proposition 7.3. If $\mathcal{I}$ satisfies E, $\mathcal{I}$ satisfies E^Γ and $E^\Gamma \in \mathcal{D}^\Gamma(\mathcal{R}C;\Diamond^* t)$. Hence $\delta_{\mathcal{I}} Dt \le \delta_{\mathcal{I}} E^\Gamma t$. The claim follows by Proposition 7.4.

   We show $\delta_{\mathcal{I}} Dt > 0$. For contradiction, let $\delta_{\mathcal{I}} Dt = 0$. Then $\mathcal{I}, a \vDash D;t$ for some a. Thus $\mathcal{I}, a \vDash \mathcal{R}C;t$ by Proposition 3.4 (1). Contradiction. ∎

We have now arrived at a decision procedure for the sublogic H^*.

8 Tableaux for H^* with @

It is straightforward to extend our results to the full logic $\mathrm{H}^*_@$, which adds formulas of the form $@_x s$ to H^*. A quasi-branch Γ **realizes a literal** $@_x s$ if it contains a clause D such that $D \rhd \{x, s\}$. A branch Γ is **evident** if it is evident as defined for H^* and in addition realizes every literal $@_x s$ such that $@_x s \in C$ for some clause $C \in \Gamma$. It is easy to verify that the model existence theorem for H^* extends to the full logic $\mathrm{H}^*_@$. The realization condition for @ leads to an additional expansion rule.

Expansion Rule for @
If $@_x s \in C \in C\Gamma$ and Γ does not realize $@_x s$,
then expand Γ to all branches $\Gamma;D$ such that $D \in \mathcal{D}^\Gamma\{x, s\}$.

Since the new rule respects the subterm closure, termination is preserved. The soundness of the new rule is easy to show.

Proposition 8.1 (Soundness of Rule for @) Let $\mathcal{I}$ be a straight model of a branch Γ and let $@_x s \in C \in \Gamma$. Then there exists a clause $D \in \mathcal{D}^\Gamma\{x, s\}$ such that $\Gamma;D$ is a branch and $\mathcal{I}$ is a straight model of $\Gamma;D$.

9 Optimizations

We give two additional rules that realize certain diamond literals with links to already present clauses, thus avoiding the introduction of unnecessary clauses and unnecessary branchings. This way the size of the tableaux the decision procedure has to explore can be reduced.

Additional Expansion Rule for Simple Diamonds
If $\Diamond s \in C \in \mathcal{C}\Gamma$ and Γ does not realize $\Diamond s$ in C
and $\Diamond s$ is simple and $D \in \Gamma$ and $D \rhd \mathcal{R}C\,;s$,
then expand Γ to $\Gamma\,; C(\Diamond s)D$.

Additional Expansion Rule for Eventualities
If $\Diamond^+ s \in C \in \mathcal{C}\Gamma$ and Γ does not realize $\Diamond^+ s$ in C
and $D \in \Gamma$ and $D \rhd \mathcal{R}C\,;s$,
then expand Γ to $\Gamma\,; C(\Diamond^+ s)D$.

Both rules preserve straight models and yield extensions that are branches. This suffices for their correctness.

A branch Γ is **quasi-evident** if there is some set Δ of links such that $\Gamma \cup \Delta$ is an evident branch. It suffices if the decision procedure stops with quasi-evident branches rather than evident branches. This provides for optimizations since it allows the decision procedure not to introduce new clauses and branchings for diamond formulas that can be realized with links to existing clauses. The two expansion rules given above are subsumed by this optimization.

Let Γ be a branch. A clause C is **subsumed in Γ** if C contains no eventualities and there is a clause $D \in \Gamma$ such that $C \subsetneq D$. There is no need to realize literals in subsumed clauses.

Proposition 9.1 A branch that realizes all literals of the form $\Diamond s$ or $@_x s$ in non-subsumed clauses is quasi-evident and hence has a finite model.

The left branch in Fig. 1 explains why subsumed clauses must not contain eventualities.

Finally, we remark that bad-loop checking can be done in quasi-constant time when a branch is extended with a new link. For this, one maintains a data structure that for every clause C and every unrealized eventuality $\Diamond^+ s \in C$ provides all clauses D such that a link $C(\Diamond^+ s)D$ would result in a bad loop. Such a data structure can be maintained in quasi-constant time. History variables as used in De Giacomo and Massacci [8] are one possibility to realize such a data structure.

10 Conclusion

This paper presents the first terminating tableau system for hybrid logic with eventualities. The system employs a novel clausal form that abstracts away from propositional reasoning. In contrast to existing systems for hybrid logic, the system presented here does not employ prefixes.

We are interested in extensions of the system. One interesting extension are difference modalities [17, 18], which introduce equational constraints that seem to require the introduction of fresh nominals. Another interesting extension are converse modalities. Finally, there is the extension to hybrid PDL.

Another interesting direction for future work is the implementation of a prover based on the system presented in this paper and to compare its performance to existing provers for PDL [13, 12] on benchmarks with eventualities and to our hybrid logic prover Spartacus [14] on benchmarks for modal and hybrid logic.

References

[1] Pietro Abate, Rajeev Goré, and Florian Widmann. An on-the-fly tableau-based decision procedure for PDL-satisfiability. In Carlos Areces and Stéphane Demri, editors, *Proc. 5th Workshop on Methods for Modalities (M4M-5)*, volume 231 of *Electr. Notes Theor. Comput. Sci.*, pages 191–209. Elsevier, 2009.

[2] Carlos Areces and Balder ten Cate. Hybrid logics. In Patrick Blackburn, Johan van Benthem, and Frank Wolter, editors, *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*, pages 821–868. Elsevier, 2007.

[3] Franz Baader. Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. Technical Report RR-90-13, DFKI, 1990.

[4] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge University Press, 2001.

[5] Thomas Bolander and Patrick Blackburn. Termination for hybrid tableaus. *J. Log. Comput.*, 17(3):517–554, 2007.

[6] Thomas Bolander and Torben Braüner, editors. *M4M-6*, number 128 in Computer Science Research Reports, Roskilde, Denmark, 2009. Roskilde University. To appear in ENTCS.

[7] Thomas Bolander and Torben Braüner. Tableau-based decision procedures for hybrid logic. *J. Log. Comput.*, 16(6):737–763, 2006.

[8] Giuseppe De Giacomo and Fabio Massacci. Combining deduction and model checking into tableaux and algorithms for converse-PDL. *Inf. Comput.*, 162(1–2):117–137, 2000.

[9] E. Allen Emerson and Edmund M. Clarke. Using branching time temporal logic to synthesize synchronization skeletons. *Sci. Comput. Programming*, 2(3):241–266, 1982.

[10] E. Allen Emerson and Joseph Y. Halpern. "Sometimes" and "not never" revisited: On branching versus linear time temporal logic. *J. ACM*, 33(1):151–178, 1986.

[11] Michael J. Fischer and Richard E. Ladner. Propositional dynamic logic of regular programs. *J. Comput. System Sci.*, pages 194–211, 1979.

[12] Oliver Friedmann and Martin Lange. A solver for modal fixpoint logics. In Bolander and Braüner [6], pages 176–187. To appear in ENTCS.

[13] Rajeev Goré and Florian Widmann. An optimal on-the-fly tableau-based decision procedure for PDL-satisfiability. In Renate A. Schmidt, editor, *CADE 2009*, volume 5663 of *LNCS*, pages 437–452. Springer, 2009.

[14] Daniel Götzmann, Mark Kaminski, and Gert Smolka. Spartacus: A tableau prover for hybrid logic. In Bolander and Braüner [6], pages 201–212. To appear in ENTCS.

[15] David Harel, Dexter Kozen, and Jerzy Tiuryn. *Dynamic Logic*. The MIT Press, 2000.

[16] Ian Horrocks and Ulrike Sattler. A tableau decision procedure for $\mathcal{SHOIQ}$. *J. Autom. Reasoning*, 39(3):249–276, 2007.

[17] Mark Kaminski and Gert Smolka. Terminating tableaux for hybrid logic with the difference modality and converse. In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *IJCAR 2008*, volume 5195 of *LNCS*, pages 210–225. Springer, 2008.

[18] Mark Kaminski and Gert Smolka. Terminating tableau systems for hybrid logic with difference and converse. *J. Log. Lang. Inf.*, 18(4):437–464, 2009.

[19] Amir Pnueli. The temporal logic of programs. In *Proc. 18th Annual Symp. on Foundations of Computer Science (FOCS'77)*, pages 46–57. IEEE Computer Society Press, 1977.

[20] Vaughan R. Pratt. A near-optimal method for reasoning about action. *J. Comput. System Sci.*, 20(2):231–254, 1980.

[21] Ulrike Sattler and Moshe Y. Vardi. The hybrid μ-calculus. In Rajeev Goré, Alexander Leitsch, and Tobias Nipkow, editors, *IJCAR 2001*, volume 2083 of *LNCS*, pages 76–91. Springer, 2001.

[22] Sigurd Schneider. *Terminating Tableaux for Modal Logic with Transitive Closure*. Bachelor's thesis, Saarland University, 2009.