

Tableau-based Decision Procedure for Hybrid Logic with Satisfaction Operators, Universal Modality and Difference Modality

Michał Zawidzki^{1,2}

¹ Department of Logic, University of Lodz
16/18 Kopcińskiego St., 90-232 Lodz, Poland

² School of Computer Science, The University of Manchester
Oxford Road, Manchester M13 9PL, United Kingdom
michal.zawidzki@gmail.com

Abstract. Hybrid logics are extensions of standard modal logics, which significantly increase the expressive power of the latter. Since most of hybrid logics are known to be decidable, decision procedures for them is a widely investigated field of research. So far, several tableau calculi for hybrid logics have been presented in the literature. In this paper we introduce a sound, complete and terminating tableau calculus $\mathcal{T}_{\mathcal{H}(\@,E,D)}$ for hybrid logics with satisfaction operators, universal modality and difference modality. $\mathcal{T}_{\mathcal{H}(\@,E,D)}$ not only uniformly covers relatively wide range of various hybrid logics but it is also conceptually simple and enables effective search for a minimal model for a satisfiable formula. $\mathcal{T}_{\mathcal{H}(\@,E,D)}$ exploits the *unrestricted blocking* mechanism introduced as an explicit, sound tableau rule.

Keywords: hybrid logics, modal logics, tableau algorithms, decision procedures, automated reasoning

1 Introduction

Hybrid logics are powerful extensions of modal logics which allow referring to particular states of a model without using meta-language. In order to achieve it, the language of standard modal logics is enriched with the countably infinite set of propositional expressions called *nominals* (we fix the notation $\text{NOM} = \{i, j, k, \dots\}$ to stand for the set of nominals), disjoint from the set of propositional variables PROP . Each nominal is true at exactly one world and therefore can serve both as a label and as a formula. Supplying a language with nominals significantly strengthens its expressive power. In the presented paper we also consider further modifications of hybrid logic obtained by adding the so-called satisfaction operators, the universal modality and the difference modality. The satisfaction operators of the form $@_i$ allow stating that a particular formula holds at a world labelled by i . The universal modality E expresses the fact that there exists a world in a domain, at which a particular formula holds. The difference modality D stands for the fact that a particular formula holds at a world different from the current one.

Some hybrid logics additionally contain a different sort of expressions, the *state variables*, which allow quantifying over worlds, and additional operators like the *down-arrow operator* or the *state quantifiers*. However, these logics are proven to be undecidable (cf. [1]) so, in principle, they cannot be subjected to a terminating tableau-based decision procedure. We therefore confine ourselves only to the forgoing decidable hybrid logic.

In the present paper we introduce a sound, complete and terminating tableau calculus $\mathcal{T}_{\mathcal{H}(\@, E, D)}$ for hybrid logics with @, E and D operators. Our approach, unlike that in [7] and [3], is focused on the uniform treatment of all aforementioned logics, conceptual simplicity and minimality of models generated by $\mathcal{T}_{\mathcal{H}(\@, E, D)}$. Basing on [10], we introduce the *unrestricted blocking* mechanism that satisfies these conditions.

In Section 2 a characterisation of the logic $\mathcal{H}(\@, E, D)$ is provided. In Section 3 we introduce the tableau calculus $\mathcal{T}_{\mathcal{H}(\@, E, D)}$ and we describe the decision procedure for $\mathcal{H}(\@, E, D)$. In Section 4 we prove soundness and completeness of $\mathcal{T}_{\mathcal{H}(\@, E, D)}$ and Section 5 provides a closer look at the termination problem for $\mathcal{T}_{\mathcal{H}(\@, E, D)}$. We conclude the paper in Section 6.

2 Hybrid logic

Syntax

Let $\mathcal{O} \in \{\@, E, D\}$. By $\mathcal{H}(\mathcal{O})$ we will denote the hybrid logic with operator(s) \mathcal{O} .

We recursively define the set FORM of well-formed formulas of the logic $\mathcal{H}(\@, E, D)$ in the following manner:

$$\text{FORM} \ni \varphi := p \mid i \mid \neg\psi \mid \psi \wedge \chi \mid \diamond\psi \mid \@_i\psi \mid E\psi \mid D\psi,$$

where $p \in \text{PROP}$, $i \in \text{NOM}$ and $\psi, \chi \in \text{FORM}$.

Other connectives and operators are defined in a standard way. Both E and D have dual operators. @ is self-dual. We abbreviate $\neg E \neg$ as A.

Semantics

A model \mathfrak{M} for hybrid logic $\mathcal{H}(\@, E, D)$ is a triple $\langle W, R, V \rangle$ where:

- $W \neq \emptyset$ is called a *domain*,
- $R \subseteq W^2$ is called an *accessibility relation*,
- $V : \text{PROP} \cup \text{NOM} \rightarrow \mathcal{P}(W)$ such that for each $i \in \text{NOM}$ $V(i)$ is a singleton set; V is called a *valuation function*.

Relation \models (*forcing*) is defined inductively:

$$\begin{array}{lll} \mathfrak{M}, w \models p & \Leftrightarrow & w \in V(p), \quad p \in \text{PROP}; \\ \mathfrak{M}, w \models i & \Leftrightarrow & \{w\} = V(i), \quad i \in \text{NOM}; \\ \mathfrak{M}, w \models \neg\varphi & \Leftrightarrow & \mathfrak{M}, w \not\models \varphi; \\ \mathfrak{M}, w \models \varphi \wedge \psi & \Leftrightarrow & \mathfrak{M}, w \models \varphi \wedge \mathfrak{M}, w \models \psi; \\ \mathfrak{M}, w \models \diamond\varphi & \Leftrightarrow & \exists z \in W (wRz \wedge \mathfrak{M}, z \models \varphi); \\ \mathfrak{M}, w \models \@_i\varphi & \Leftrightarrow & \{z\} = v(i) \wedge \mathfrak{M}, z \models \varphi; \\ \mathfrak{M}, w \models E\varphi & \Leftrightarrow & \exists z \in W (\mathfrak{M}, z \models \varphi); \\ \mathfrak{M}, w \models D\varphi & \Leftrightarrow & \exists z \in W (z \neq w \wedge \mathfrak{M}, z \models \varphi), \end{array}$$

where $\Leftrightarrow, \exists, \dot{\wedge}, \models, \dot{=}$ are meta-language symbols. Henceforth, we will call the expressions containing these meta-language symbols the *domain expressions*.

3 Synthesising tableau calculus for the logic $\mathcal{H}(@, \mathbf{E}, \mathbf{D})$

Tableau calculi

Two main types of tableau calculi for hybrid logics are present in the literature, namely the *prefixed* and the *internalised* calculi. The prefixed calculi consist in introducing another sort of expressions, namely *prefixes*. They serve as labels for worlds, which, unlike nominals, are of meta-linguistic provenience. Another type of meta-language expressions occurring in prefixed tableaux are the *accessibility expressions*. The equality between two prefixes is expressed implicitly by imposing on them the satisfaction of the same nominal. Apparently, prefixed calculi are less complex than internalised calculi. Besides, basic hybrid logic \mathcal{H} is not supplied with sufficient expressive power to internalise its own semantics. It therefore requires the domain expressions occurring in the calculus. The most widely known prefixed tableau calculi for hybrid logics come from Tzakova [12], Bolander and Braüner (who improved Tzakova's calculus to the terminating version) [5], Kaminski and Smolka [8]. The tableau calculus for hybrid logics obtained than the synthesised framework from [10] is also subsumed under the prefixed calculi class.

Internalised calculi for hybrid logics take advantage of the high expressive power of these logics which allows encoding the domain expressions within the language. Although internalisation of the logic allows dispensing with certain rules present in prefixed tableau calculi, it also jeopardises termination of the calculus by, e.g., using pure axioms (not including other formulas but nominals) to characterise frame conditions (cf. [3]).

In this section we present an internalised tableau calculus covering hybrid logics with the satisfaction operators, the universal modality and the difference modality. It resembles Blackburn's calculus from [2] modified by Bolander and Braüner in [5] and by Blackburn and Bolander in [3]. However, certain rules have been added (e.g. the rules for D).

Encoding the domain expressions

In [2] Blackburn made an observation that the language of hybrid logic with @ operators is sufficiently rich to express semantics within itself. As we mentioned in Section 2, there are three types of the domain expressions: satisfaction statements ($\mathfrak{M}, w \models \varphi$), accessibility statements (wRv) and equality statements ($w \dot{=} v$). Hybrid equivalents of the forgoing expressions are shown below.

$$\begin{aligned} \mathcal{H}(\mathfrak{M}, w \models \varphi) &= @_{i_w} \varphi & \mathcal{H}(\mathfrak{M}, w \not\models \varphi) &= @_{i_w} \neg \varphi \\ \mathcal{H}(R(w, v)) &= @_{i_w} \diamond j_v & \mathcal{H}(\neg R(w, v)) &= @_{i_w} \neg \diamond j_v \\ \mathcal{H}(w \dot{=} v) &= @_{i_w} j_v & \mathcal{H}(w \dot{\neq} v) &= @_{i_x} \neg j_y \end{aligned}$$

Both E and D operators allow mimicking @ operators: $@_i \varphi := \mathbf{E}(i \wedge \varphi)$ and $@_i \varphi := (i \wedge \varphi) \vee \mathbf{D}(i \wedge \varphi)$. Therefore, in the calculus we use the notation $i : \varphi$, rather than $@_i \varphi$, to keep its universal character. This *colon* notation will stand for one of the forgoing expressions, depending on a considered logic, except for the fact that whenever a logic includes @ operators, $i : \varphi$ means $@_i \varphi$.

Rules for the connectives:

$$\begin{array}{cccc}
(\neg) \frac{i : \neg j}{j : j} & (\neg\neg) \frac{i : \neg\neg\varphi}{i : \varphi} & (\wedge) \frac{i : \varphi \wedge \psi}{i : \varphi, i : \psi} & (\neg\wedge) \frac{i : \neg(\varphi \wedge \psi)}{i : \neg\varphi \mid i : \neg\psi} \\
(\diamond)^* \frac{i : \diamond\varphi}{i : \diamond j, j : \varphi} & (\neg\diamond) \frac{i : \neg\diamond\varphi, i : \diamond j}{j : \neg\varphi} & (@) \frac{i : @_j\varphi}{j : \varphi} & (\neg@) \frac{i : \neg@_j\varphi}{j : \neg\varphi} \\
(E)^* \frac{i : E\varphi}{j : \varphi} & (\neg E) \frac{i : \neg E\varphi, j : j}{j : \neg\varphi} & (D)^* \frac{i : D\varphi}{i : \neg j, j : \varphi} & (\neg D) \frac{i : \neg D\varphi, j : j}{i : j \mid j : \neg\varphi}
\end{array}$$

Equality rules:

$$(\text{ref}) \frac{i : \varphi}{i : i} \quad (\text{sub}) \frac{i : j, i : \varphi}{j : \varphi}$$

Closure rule and unrestricted blocking rule:

$$(\perp) \frac{i : \varphi, i : \neg\varphi}{\perp} \quad (\text{ub}) \frac{i : i, j : j}{i : j \mid i : \neg j}$$

* Nominals in the conclusions are fresh on the branch.

Fig. 1. Rules for the calculus $\mathcal{T}_{\mathcal{H}(\@, E, D)}$

Tableau calculus

Figure 1 presents the rules of the tableau calculus $\mathcal{T}_{\mathcal{H}(\@, E, D)}$ for the logic $\mathcal{H}(\@, E, D)$.

Boolean rules are straightforward and require no additional comments. (\diamond) , (E) and (D) are rules introducing new labels, which was marked as the side-condition for them. In the case of $(\neg E)$ and $(\neg D)$ the standard side-condition of former occurrence of a label on a branch was replaced by introducing an explicit premiss stating that a particular nominal has appeared as a label on a branch. The rule (ref) is a reflexivity rule that introduces to a branch the explicit information that a nominal occurred as a label within a branch. (sub) expresses the substitutability of two nominals as labels, provided that one of them is labelled by the other. The (\perp) rule is self-evident. The (ub) rule is a variant of the *analytical cut rule* applied to nominals. Intuitively, if two labels appear on a branch, they either label two distinct worlds or the same world. Thus, (ub) allows comparing any pair of labels that appeared on a branch. As it will turn out before long, this possibility is essential for termination of the whole calculus. The rule $(\neg D)$ deserves a wider comment. In [3] Blackburn and Bolander notice that the $(\neg D)$ rule of the form

$$\frac{i : \neg D\varphi, i : \neg j}{j : \neg\varphi}$$

breaks the completeness of the whole calculus. In [8] Kaminski and Smolka formulate $(\neg D)$ correctly but they do not explain why no such modification like from [3] can be applied to $(\neg D)$. In [10] Schmidt and Tishkovsky introduce the (\dagger) condition which decides whether the refinement can be applied to a rule:

Theorem 1 ([10]). *Let \mathcal{T} be a tableau calculus. Let β be the rule of the form $\frac{X_0}{X_1 \mid \dots \mid X_n}$ and let \mathcal{T}^R be a refined version of \mathcal{T} . Suppose that \mathcal{B} is an arbitrary open and fully-expanded branch in a \mathcal{T}^R -tableau. Let $F = \{\varphi_1, \dots, \varphi_l\}$ be a set of all $\mathbf{K}(E_n)$ -formulas from \mathcal{B} reflected in $\mathfrak{M}(\mathcal{B})$. Then the refined rule $\frac{X_0, \neg X_{i_1}, \dots, \neg X_{i_j}}{X_{i_{j+1}} \mid \dots \mid X_{i_n}}$ is admissible (i.e. the*

resulting calculus \mathcal{T}^R is still complete) satisfies the following condition is satisfied:

$$\begin{aligned} & \text{If } X_0(\varphi_{i_1}, \dots, \varphi_{i_k}) \in \mathcal{B} \\ & \text{then } \mathfrak{M}(\mathcal{B}) \models X_m(\varphi_{i_1}, \dots, \varphi_{i_k}), \text{ for some } m \in \{1, \dots, n\}. \end{aligned} \quad (\dagger)$$

(\dagger) holds for $(\neg\Diamond)$ in most modal and description logics but, as it turns out, it fails for $(\neg D)$.

Before we provide a proper method of constructing a tableau, we need to introduce several preliminary definitions.

Definition 1. We call a branch of a $\mathcal{T}_{\mathcal{H}(\@,E,D)}$ tableau closed if the closure rule was applied on it. If a branch is not closed, it is open. An open branch is fully expanded if no other rules are applicable on it.

Definition 2. Let $\text{NOM}(\mathcal{B})$ be a set of nominals occurring as labels on a fully expanded branch \mathcal{B} of a $\mathcal{T}_{\mathcal{H}(\@,E,D)}$ tableau for a given input formula. We introduce the $\leftrightarrow_{\mathcal{B}}$ relation over $\text{NOM}(\mathcal{B})$ which we define in the following way:

$$i \leftrightarrow_{\mathcal{B}} j \text{ iff } i : j \in \mathcal{B}.$$

Proposition 1. $\leftrightarrow_{\mathcal{B}}$ is the equivalence relation.

Proof. Reflexivity is ensured by the (ref) rule. For symmetry assume that $i : j$ is on \mathcal{B} . By (ref) we obtain $i : i$ and after applying (sub) to these two premises we obtain $j : i$. For transitivity suppose that $i : j$ and $j : k$ are on \mathcal{B} . By symmetry we have that $j : i$ is also on \mathcal{B} . We therefore take $j : i$ and $j : k$ as premises of (sub) and obtain $i : k$.

Definition 3. A rule of the $\mathcal{T}_{\mathcal{H}(\@,E,D)}$ is applied eagerly in a tableau iff whenever it is applicable, it is applied.

Definition 4. Let $\prec_{\mathcal{B}}$ be an ordering on $\text{NOM}(\mathcal{B})$ defined as follows:

$$i \prec_{\mathcal{B}} j \text{ iff } i : i \text{ occurred on } \mathcal{B} \text{ earlier than } j : j.$$

Note that $\prec_{\mathcal{B}}$ is well-founded and linear since no rule introduces more than 1 labelling nominal as a conclusion.

Definition 5. To each $\mathcal{T}_{\mathcal{H}(\@,E,D)}$ rule we affix the priority number. It indicates what the order of application of particular rules should be. The lower the number is, the sooner the rule should be applied. We have: (ref), (\neg) : 1, (ub): 2, (sub): 3, $(\neg\neg)$, (\wedge) , $(\neg\wedge)$: 4, $(\neg\Diamond)$, $(\neg E)$, $(\neg D)$: 5, (\Diamond) , (E) , (D) : 6.

Now we are ready to provide the tableau construction algorithm. As usual, we do it inductively.

Definition 6 (Tableau construction algorithm). *Basic step:* For a given input formula φ put $i : \neg\varphi$ at the initial node. i is a nominal not occurring in φ .

Inductive step: Suppose that you performed n steps of a derivation. In the $n+1$ th step apply the rules of $\mathcal{T}_{\mathcal{H}(\@,E,D)}$ eagerly respecting the priority ordering given in Definition 5 and fulfilling the following conditions:

- (c1) if the application of a rule results in a formula that is already present on a branch, do not perform this application;

- (c2) rules of priority 5 and 6 can only be applied to labels that are the least elements (with respect to $\prec_{\mathcal{B}}$) of the equivalence class (with respect to $\rightsquigarrow_{\mathcal{B}}$);
- (c3) the (\diamond) must not be applied to formulas of the form $i : \diamond j$. We call them the accessibility formulas;
- (c4) apply the (\perp) rule whenever it is possible.

If after the $n + 1$ th step of derivation:

- (a) all tableau branches are closed, stop and return: theorem,
- (b) there are open branches in a tableau and no further rules are applicable (respecting conditions (c1)-(c4)), stop and return: non-theorem;
- (c) there are open branches in a tableau and further rules are applicable (respecting conditions (c1)-(c4)), proceed to the $n + 2$ th step.

We will explain the way the (ub) rule works more carefully in Section 5.

4 Soundness and Completeness of $\mathcal{T}_{\mathcal{H}(\@, \mathbf{E}, \mathbf{D})}$

In the current section we state and prove soundness and completeness of the forgoing calculus. First, we formulate the following

Definition 7. We call a tableau calculus \mathcal{T} sound if and only if for each satisfiable input formula φ each tableau $\mathcal{T}(\varphi)$ is open, i.e., there exists a fully expanded branch on which no closure rule was applied. A tableau calculus is called complete if and only if for each unsatisfiable input formula φ there exists a closed tableau, i.e. a tableau where a closure rule was applied on each branch.

For soundness it amounts to proving that particular rules preserve satisfiability. For completeness we take the contrapositive of the condition given in Definition 7 and demonstrate that if there exists an open, fully expanded branch \mathcal{B} in a tableau for φ then there exists a model for φ .

Theorem 2. $\mathcal{T}_{\mathcal{H}(\@, \mathbf{E}, \mathbf{D})}$ is sound.

Proof. By easy verification of all the rules.

Suppose that \mathcal{B} is an open, fully expanded branch in a $\mathcal{T}_{\mathcal{H}(\@, \mathbf{E}, \mathbf{D})}$ tableau for φ . We define a model $\mathfrak{M}(\mathcal{B}) = \langle W, R, V \rangle$ derived from \mathcal{B} in the following way:

$$\begin{aligned} W &= \{[i]_{\rightsquigarrow_{\mathcal{B}}} \mid i : i \in \mathcal{B}\}; \\ R &= \{([i]_{\rightsquigarrow_{\mathcal{B}}}, [j]_{\rightsquigarrow_{\mathcal{B}}}) \mid i : \diamond j \in \mathcal{B}\}; \\ V &= \{(i, [i]_{\rightsquigarrow_{\mathcal{B}}}) \mid i : i \in \mathcal{B}\} \cup \{(p, U) \mid p \in \text{PROP}, p \text{ occurred in } \mathcal{B} \text{ and } U = \{[i]_{\rightsquigarrow_{\mathcal{B}}} \mid i : p \in \mathcal{B}\}\}. \end{aligned}$$

Lemma 1. Suppose that \mathcal{B} is an open, fully expanded branch in a $\mathcal{T}_{\mathcal{H}(\@, \mathbf{E}, \mathbf{D})}$ tableau for φ . Then if $i : \psi \in \mathcal{B}$ then $\mathfrak{M}(\mathcal{B}), [i]_{\rightsquigarrow_{\mathcal{B}}} \models \psi$.

Proof. By induction on the complexity of ψ . Since all cases save $\psi = \mathbf{D}\chi$ and $\psi = \neg\mathbf{D}\chi$ are covered by proofs given in [3] and [5], we only consider missing cases.

Case: $\psi = \mathbf{D}\chi$. We have $i : \mathbf{D}\chi \in \mathcal{B}$. After applying (D) we obtain $i : \neg j \in \mathcal{B}$ and $j : \chi \in \mathcal{B}$. By the inductive hypothesis we have that $\mathfrak{M}(\mathcal{B}), [j]_{\rightsquigarrow_{\mathcal{B}}} \models \chi$. It suffices to show that $[i]_{\rightsquigarrow_{\mathcal{B}}}$ and $[j]_{\rightsquigarrow_{\mathcal{B}}}$ are distinct. Suppose that they are the same equivalence class. But then, by Def. 2, $i : j \in \mathcal{B}$, which contradicts the fact that \mathcal{B} is open.

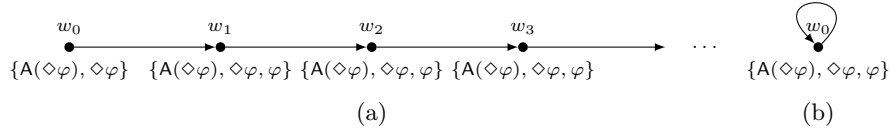


Fig. 2. (a) and (b) present, respectively, an infinite and a finite (minimal) model for the formula $A(\diamond\varphi)$. Both of them can be obtained from a tableau if the (ub) rule is involved, since it allows merging worlds in an arbitrary way, provided that the consistency is preserved.

Case: $\psi = \neg D\chi$. We have $i : \neg D\chi \in \mathcal{B}$. If no labels different than i occurred on \mathcal{B} , it means that $W = \{[i]_{\leftrightarrow_{\mathcal{B}}}\}$ and therefore $\mathfrak{M}(\mathcal{B}), [i]_{\leftrightarrow_{\mathcal{B}}} \models \neg D\chi$ trivially holds. Suppose that \mathcal{L} is a set of labels different than i , which occurred in \mathcal{B} . Pick an arbitrary label j from \mathcal{L} . After applying ($\neg D$) to $\neg D\chi$ we obtain that either $i : j \in \mathcal{B}$ or $j : \neg\chi \in \mathcal{B}$. If the former is the case, by the inductive hypothesis we obtain that $[i]_{\leftrightarrow_{\mathcal{B}}} = [j]_{\leftrightarrow_{\mathcal{B}}}$. If the latter holds, then by the inductive hypothesis, $\mathfrak{M}(\mathcal{B}), [j]_{\leftrightarrow_{\mathcal{B}}} \models \neg\chi$. Both cases are subsumed by $\mathfrak{M}(\mathcal{B}), [i]_{\leftrightarrow_{\mathcal{B}}} \models \neg D\chi$. Since j was picked arbitrarily, we obtain the conclusion.

Theorem 3. $\mathcal{J}_{\mathcal{H}(\@, E, D)}$ is complete.

Proof. By Definition 7 and Lemma 1.

5 Termination of $\mathcal{J}_{\mathcal{H}(\@, E, D)}$

Exploiting the (ub) rule and the conditions (c1)-(c4) we show that $\mathcal{J}_{\mathcal{H}(\@, E, D)}$ is terminating for the logic $\mathcal{H}(\@, E, D)$, provided that it has the *finite model property* for a certain class of frames.

First, we make a remark that will be useful afterwards (cf. [11]).

Remark 1. For each $[i]_{\leftrightarrow_{\mathcal{B}}}$ the number of applications of the rules introducing a new label, namely (\diamond), (E), (D), to members of $[i]_{\leftrightarrow_{\mathcal{B}}}$ is finite.

Proof. Indeed, if the (ub) is eagerly applied and the conditions (c2) and (c3) are fulfilled, it ensures that no superfluous application of (\diamond), (E), (D) is performed, since they are only applied to one member of $[i]_{\leftrightarrow_{\mathcal{B}}}$ and are not applied to accessibility formulas (otherwise it would lead to an infinite derivation that could not be subjected to blocking). Since the input formula φ is assumed to be finite, therefore for each i that occurred in \mathcal{B} $[i]_{\leftrightarrow_{\mathcal{B}}}$ the number of (\diamond), (E), (D) applications is finite.

Corollary 1. For each $\mathcal{J}_{\mathcal{H}(\@, E, D)}$ tableau branch \mathcal{B} is finite iff W of $\mathfrak{M}(\mathcal{B})$ is finite.

Now we are ready to state the lemma that is essential for termination of $\mathcal{J}_{\mathcal{H}(\@, E, D)}$. However, before we do this, we explain informally how the (ub) rule works. Our tableau calculus by default handles all distinct nominals that were introduced to a branch as labelling distinct worlds. It leads to a situation where a satisfiable formula having a simple model generates an infinite tableau (see Fig. 2). The (ub) rule compares all labels that occurred in a branch and its left conclusion merges each pair unless it leads to the

inconsistency. As a consequence, if a formula has a model \mathfrak{M} of a certain cardinality, it will be reflected by a finite, fully expanded open branch of a $\mathcal{T}_{\mathcal{H}(\textcircled{\text{A}}, \text{E}, \text{D})}$ tableau. The reason is that the left conclusion of the (ub) rule decreases the cardinality of a model whenever possible, so a model of the cardinality not-greater than the cardinality of \mathfrak{M} will eventually be obtainable from one of the branches of a tableau. The formal argument is presented in the following lemma.

Lemma 2. *Suppose that a finite model $\mathfrak{N} = \langle W', R', V' \rangle$ satisfies a formula φ . Then there exists an open branch \mathcal{B} in a $\mathcal{T}_{\mathcal{H}(\textcircled{\text{A}}, \text{E}, \text{D})}$ tableau and $\mathfrak{M}(\mathcal{B}) = \langle W, R, V \rangle$ such that $\text{Card}(W) \leq \text{Card}(W')$.*

Proof. We proceed by induction on the number of steps in the derivation. During the derivation we construct a branch \mathcal{B} in such a way that $\mathfrak{M}(\mathcal{B})$ is partially isomorphic to \mathfrak{N} (cf. [11]).

Basic step: φ is satisfiable on \mathfrak{N} , so there must exist $w \in W'$ such that $\mathfrak{N}, w \models \varphi$. If also $\mathfrak{N}, w \models i$ such that i does not occur in φ , we put at the initial node of the derivation $i : \varphi$. If no such nominal holds in w , we conservatively extend \mathfrak{N} by adding fresh nominal i to w and put at the initial node of the derivation $i : \varphi$.

Inductive step: Application of each tableau rule should be considered as a separate case. Nevertheless, only four rules seem to be essential for this proof, namely (\diamond) , (E), (D) and (ub), i.e. rules that either introduce a new label to a branch or identify labels already present on a branch. We consider each of them.

Case: (\diamond) . Suppose that a formula $\diamond\psi$ occurred at the n th node of the derivation. It means that we associated the label i of this node with a world in W' that satisfies $\diamond\psi$ and i . What follows, there must exist a world v such that wRv and $\mathfrak{N}, v \models \psi$. If v does not satisfy any nominal l that has not yet occurred on the branch either as a label or as a subformula, we conservatively extend \mathfrak{N} by ascribing l to v . Applying (\diamond) to $\diamond\psi$ we obtain $i : \diamond j$ and $j : \psi$. We put l in place of j .

Case: (E). Suppose that a formula $\text{E}\psi$ occurred at the n th node of the derivation. It means that we associated the label i of this node with a world in W' that satisfies $\text{E}\psi$ and i . Therefore, there exists a world v such that $\mathfrak{N}, v \models \psi$. If v does not satisfy any nominal l that has not yet occurred on a branch either as a label or as a subformula, we conservatively extend \mathfrak{N} by ascribing l to v . Applying (E) to $\text{E}\psi$ we obtain $j : \psi$. We put l in place of j .

Case: (D). Suppose that a formula $\text{D}\psi$ occurred at the n th node of the derivation. It means that we affixed the label i of this node to a world in W' that satisfies $\text{D}\psi$ and i . Therefore, there exists a world v such that $\mathfrak{N}, v \models \psi \wedge \neg i$. If v does not satisfy any nominal l that has not yet occurred on a branch either as a label or as a subformula, we conservatively extend \mathfrak{N} by ascribing l to v . Applying (D) to $\text{D}\psi$ we obtain $j : \neg i$ and $j : \psi$. We put l in place of j .

Case: (ub). Suppose that during the derivation two labels i and j have been introduced to \mathcal{B} . By the inductive hypothesis we mapped these labels to worlds w and v of (the conservative extension of) W' . Either the world w satisfies $i \wedge j$ (which would mean that w and v are the same world) or it satisfies $i \wedge \neg j$ (which indicates that w and v are distinct). If the former is the case, we pick the left conclusion of (ub) and add it to \mathcal{B} , if the latter is the case, we choose the right conclusion of (ub) and add it to \mathcal{B} .

Since \mathcal{B} is open, we can construct a model $\mathfrak{M}(\mathcal{B}) = \langle W, R, V \rangle$ out of it. Now we show that $\text{Card}(W) \leq \text{Card}(W')$ (we consider \mathfrak{N} as already conservatively extended in

progress of constructing \mathcal{B}). We set a function $f : W' \rightarrow W$ as follows

$$f(w) = \begin{cases} [i]_{\leftrightarrow_{\mathcal{B}}}, & \text{if there is } i : i \in \mathcal{B} \text{ such that } i \text{ was affixed} \\ & \text{to } w \text{ during the derivation} \\ \text{arbitrary element of } W, & \text{otherwise} \end{cases}$$

f is injective and if we cut it to these elements of W' to which we assigned a nominal during the derivation, it is also an isomorphism. That concludes the proof.

To conclude our considerations it is sufficient to prove that the logic $\mathcal{H}(@, E, D)$ has the finite model property. The following proposition deals with it.

Proposition 2. *The logic $\mathcal{H}(@, E, D)$ has the effective finite model property with the bounding function $\mu = 2^{\text{Card}(\text{Sub}(\varphi))+1}$, where $\text{Sub}(\varphi)$ is a set of all subformulas of a formula φ .*

Proof. We use the standard, filtration-based argument. Suppose that a formula φ is satisfied on a (possibly infinite) model $\mathfrak{M} = \langle W, R, V \rangle$. It means that there exists $w \in W$ such that $\mathfrak{M}, w \models \varphi$. We show that there exists a finite model \mathfrak{M}' that satisfies φ and whose cardinality does not exceed $2^{\text{Card}(\text{Sub}(\varphi))+1}$.

First, we set the relation $\leftrightarrow_{\varphi}$ on W in the following way:

$$w \leftrightarrow_{\varphi} v \quad \text{iff} \quad \text{for all } \psi \in \text{Sub}(\varphi) \ \mathfrak{M}, w \models \psi \text{ iff } \mathfrak{M}, v \models \psi.$$

It is straightforward that $\leftrightarrow_{\varphi}$ is the equivalence relation

Now we are ready to construct our finite model that will satisfy φ . Let $\mathfrak{M}' = \langle W', R', V' \rangle$ such that:

$$\begin{aligned} W' &= W / \leftrightarrow_{\varphi} \uplus W / \leftrightarrow_{\varphi}; \\ R' &= \{(|v|_{\leftrightarrow_{\varphi}}, |u|_{\leftrightarrow_{\varphi}}) : R(v, u)\}; \\ V'(p) &= \{|v|_{\leftrightarrow_{\varphi}} : v \in V(p)\} \quad \text{for all proposition letters in } \varphi; \\ V'(i) &= \{|v|_{\leftrightarrow_{\varphi}} : v \in V(i)\} \quad \text{for all nominals in } \varphi. \end{aligned}$$

We prove that \mathfrak{M}' satisfies φ by induction on the complexity of subformulas of φ . Since the proof for the modal part of $\mathcal{H}(@, E, D)$ is well known (cf. [4]) and the case of $\psi = i$ follows immediately from the definition of V' , we confine ourselves to proving the cases of $@_i\chi$, $E\chi$ and $D\chi$.

Case: $\psi = @_i\chi$. Suppose that $\mathfrak{M}, v \models @_i\chi$. It means that χ holds at a world u at which also i holds. This world is transformed to a singleton equivalence class $\{u\}$ in W' .

By the inductive hypothesis it follows that $\mathfrak{M}', \{u\} \models i$ and $\mathfrak{M}', \{u\} \models \chi$. Hence $\mathfrak{M}', |v| \models @_i\chi$.

Case: $\psi = E\chi$. Suppose that $\mathfrak{M}, v \models E\chi$. It means that there exists a world u at which χ holds. By the inductive hypothesis $\mathfrak{M}', |u| \models \chi$. Hence $\mathfrak{M}', |v| \models E\chi$.

Case: $\psi = D\chi$. Suppose that $\mathfrak{M}, v \models D\chi$. It means that there exists a world u different than v , at which χ holds. By the inductive hypothesis $\mathfrak{M}', |u| \models \chi$. Two complementary cases might occur. If $|v| \neq |u|$, then we obtain $\mathfrak{M}', |v| \models \chi$. If, however, $|v| = |u|$, it means that χ is also satisfied by a copy of $|v|$ that we pasted to W' at the stage of the construction of \mathfrak{M}' . Since $|v|$ and its copy are distinct, we obtain $\mathfrak{M}', |v| \models \chi$.

Observe that pasting a distinct copy of $W / \leftrightarrow_{\varphi}$ to W' is only necessary if D is involved. Therefore, in other cases the bounding function $\mu = 2^{\text{Card}(\text{Sub}(\varphi))}$.

Consequently, we obtain the following result:

Theorem 4. $\mathcal{T}_{\mathcal{H}(@,E,D)}$ is terminating.

Proof. Follows from Corollary 1, Lemma 2 and Proposition 2.

Obviously, the bounding function μ from Proposition 2 can be reduced (cf. [1, 9]), however, the main aim of this paper is not optimising the complexity of $\mathcal{T}_{\mathcal{H}(@,E,D)}$. Besides, the filtration-based argument can be easily adapted for different types of frames. Thus, we formulate the following strategy-condition for performing the derivation in $\mathcal{T}_{\mathcal{H}(@,E,D)}$:

- (tm) Expand a branch of $\mathcal{T}_{\mathcal{H}(@,E,D)}$ -tableau until the number of equivalence classes of individuals in \mathcal{B} exceeds the bound given by μ function. Then stop.

It turns our tableau calculus into a deterministic decision procedure.

6 Concluding remarks

In this paper we presented an internalised tableau-based decision procedure for the logic $\mathcal{H}(@, E, D)$. Tableau calculus $\mathcal{T}_{\mathcal{H}(@,E,D)}$ was proven to be sound, complete and terminating. In the existing literature of the subject several approaches to systematic treatment of decision procedures for hybrid logics can be found. We recall two of them. In [3] and [5] Blackburn, Bolander and Braüner provide a terminating internalised tableau-based decision procedure for the logic $\mathcal{H}(@, E)$. However, their main concern is different from ours. Their attempts are focused on tailoring a suitable tableau calculus for each logic separately. Therefore, they introduce two different blocking mechanisms, namely *subset blocking* and *equality blocking* for the logics $\mathcal{H}(@)$ and $\mathcal{H}(@, E)$ and modify the notion of *urfather* subject to a particular logic. The resulting calculus is conceptually complex but seems to avoid any superfluous performances of the rules. In [7] Götzmann, Kaminski and Smolka describe *Spartacus*, which is a tableau prover for hybrid logics with @ operators and universal modality. Thanks to the application of advanced blocking and optimisation techniques, namely *pattern-based blocking* and *lazy branching* the system is very efficient in terms of complexity.

The decision procedure introduced in this paper presents the approach which is different from the aforementioned. It introduces (**ub**) as an explicit tableau rule which is sound and, together with the conditions (c1)-(c4), ensures termination of the whole calculus. (**ub**) allows a direct comparison of every pair of labels that occurred on a branch and, therefore, subsumes any other blocking mechanisms. (**ub**) is a generic rule which means that it generates every possible configuration of labels occurring on a branch. In comparison to [3] and [7] many of these configurations are superfluous. However, the huge advantage of this approach is conceptual simplicity which allows to avoid introducing complicated strategies of searching for a pair of labels that are liable to blocking mechanism. Additionally, for each satisfiable formula φ (**ub**) ensures that a minimal model for φ (in terms of a domain size) will be generated, which cannot be guaranteed by the systems of [3] and [7]. Moreover, $\mathcal{T}_{\mathcal{H}(@,E,D)}$ provides a uniform approach to all hybrid logics mentioned in the paper and covers the case of difference modality which is omitted in [3]. A possible direction of future work may be investigating whether the applications of the (**ub**) rule can be optimised.

Acknowledgements

The research reported in this paper is a part of the project financed from the funds supplied by the National Science Centre, Poland (decision no. DEC-2011/01/N/HS1/01979).

References

1. Areces C., Blackburn P., Marx M.: A road-map on complexity for hybrid logics. Proc. of the 8th Annual Conference of the European Association for Computer Science Logic (CSL 1999), Madrid, Spain, 307–321
2. Blackburn P.: Internalizing labelled deduction. *J. Log. Comput.* 10(1), 137–168 (2000)
3. Blackburn P., Bolander T.: Termination for Hybrid Tableaus. *J. Log. Comput.* 17(3), 517–554 (2007)
4. Blackburn P., Venema Y., de Rijke M.: *Modal Logics*. Cambridge University Press (2002)
5. Bolander T., Braüner T.: Tableau-based Decision Procedures for Hybrid Logic. *J. Log. Comput.* 16, 737–763 (2006)
6. Braüner T.: *Hybrid Logic and its Proof-Theory* Springer (2011)
7. Götzmann D., Kaminski M., Smolka G.: Spartacus: A Tableau Prover for Hybrid Logic. *Electron. Notes Theor. Comput. Sci.* 262, 127–139 (2010)
8. Kaminski M., Smolka G.: Terminating Tableau Systems for Hybrid Logic with Difference and Converse. *J. Log. Lang. Inf.* 18, 437–464 (2009)
9. Mundhenk M., Schneider T.: Undecidability of Multi-modal Hybrid Logics. *Electron. Notes Theor. Comput. Sci.* 174, 29–43 (2007)
10. Schmidt, R. A., Tishkovsky, D.: Automated Synthesis of Tableau Calculi. *Log. Meth. Comput. Sci.* 7(2), 1–32 (2011)
11. Schmidt R.A., Tishkovsky D.: Using Tableau to Decide Description Logics with Full Role Negation and Identity. <http://www.mettel-prover.org/papers/ALBOid.pdf> (2012)
12. Tzakova M.: Tableau Calculi for Hybrid Logics. LNCS 1617, 278–292 (1999)