An On-the-fly Tableau-based Decision Procedure for PDL-Satisfiability

Pietro Abate[1], Rajeev Goré[1], and Florian Widmann[1 2]

[1] Computer Science Laboratory
The Australian National University
Canberra ACT 0200, Australia
[2] Logic and Computation Programme
Canberra Research Laboratory
NICTA Australia
{Pietro.Abate|Rajeev.Gore|Florian.Widmann}@anu.edu.au

Abstract. We present a tableau-based algorithm for deciding satisfiability for propositional dynamic logic (PDL) which builds a finite rooted tree with ancestor cycles and which passes extra information from children to parents to separate good cycles from bad cycles during backtracking. It is easy to implement, with potential for parallelisation, because it constructs a potential model "on the fly" by exploring each tableau branch independently. However, its worst-case behaviour is 2EXPTIME rather than EXPTIME. We give detailed proofs of soundness and completeness in an appendix and there is a prototype implementation in the TWB (`twb.rsise.anu.edu.au`).

1 Introduction

Propositional dynamic logic (PDL) was developed in the late 1970s as a logic for reasoning about programs [15, 8, 3]. The formulae of PDL consist of traditional Boolean formulae plus "action modalities" built from a finite set of atomic programs using the operations of sequential composition ($;$), non-deterministic choice ($\cup$), and test ($?$). The satisfiability problem for PDL is known to be EXPTIME-complete [16]. Unlike the situation in EXPTIME-complete description logics, where algorithms with good average-case behaviour are known, we know of no decision procedures for testing PDL-satisfiability which are satisfactory from both a theoretical (soundness and completeness) and practical (average case behaviour) viewpoint as we explain below.

The earliest decision procedures for PDL are due to Fischer and Ladner [8] and Pratt [16]. Fischer and Ladner's method is impractical because it first constructs the set of all consistent subsets of the set of all subformulae of the given formula which always requires exponential time in all cases. On the other hand, Pratt [16] essentially builds a multi-pass (explained shortly) tableau method. Most subsequent decision procedures for other fix-point logics like propositional linear temporal logic (PLTL) [19], computation tree logic (CTL) [4, 7] and the

modal μ-calculus [14] trace their origins back to Pratt [16], and they all share one main disadvantage as explained next.

These multi-pass procedures can be described informally as follows where a "state" is a node which contains only diamond-like-formulae ("eventualities"), box-like–formulae, atoms and negated atoms. The first pass constructs a rooted tableau of nodes containing formula-sets, but allows cross-branch arcs from a state n on one branch to a (previously constructed) state m on a different branch if applying the tableau construction to n would duplicate m. Thus the first pass constructs a "pseudo-model" which is a potentially exponential-sized cyclic graph (rather than a cyclic tree where m would have to be an ancestor of n). The subsequent passes check that the "pseudo-model" is a real model by pruning inconsistent nodes and pruning nodes that contain "unfulfilled eventualities".

Although efficient model-checking techniques can check the "pseudo-model" in time which is linear in its size, these multi-pass methods can construct an exponential-sized cyclic graph when it is not necessary. One solution is to check for fulfilled eventualities "on the fly", as the graph is being built, and although such methods exist for model-checking [6, 5], we know of no such decision procedures for PDL (or for other fix-point logics). The only implementation of a multiple-pass method for PDL that we know of is in LoTRec (`www.irit.fr/Lotrec/`) but it is not optimal since it treats disjunctions in a naive way.

In 1990, Baader [2] gave a single-pass tableau-based decision procedure for a description logic with role definitions involving union, composition and transitive closure of roles which can be viewed as essentially PDL without test. His method first constructs a (cyclic tree) tableau which implements the intended semantics of the various PDL operators. To separate "good cycles" from "bad cycles", Baader's method has to decide equality of regular languages, a PSPACE-complete problem which in practice may require exponential time. Instead of solving these problems "on the fly", they can be reduced to a simple check on the identity of states in a deterministic minimal automaton created from the positive regular expressions appearing in the initial formula during a pre-processing stage [2, page 27]. But since the pre-computed automaton can be of exponential size, this alternative may require exponential time needlessly. Because Baader does not allow cross-branch cycles, his method is double-exponential in the worst-case. Finally, the "test" construct, which is essential to express "while" loops, creates a mutual recursion between the Boolean language and the regular language over programs, so it is not obvious to us how to extend Baader's method to handle "test". We know of no satisfactory implementation of this method.

In 2000, De Giacomo and Massacci [9] gave an optimal method for deciding PDL-satisfiability using labelled formulae of the form $\sigma : \varphi$ allowing them to capture the notion that "possible world σ makes formula φ true". They first give a NEXPTIME algorithm for deciding PDL-satisfiability in detail and then discuss how to obtain an EXPTIME algorithm using various known results from the literature. They do not give an actual EXPTIME algorithm, nor its soundness and completeness proofs. An initial implementation of a deterministic version of this

NEXPTIME algorithm by Schmidt and Tishkovsky struck problems for formulae with nested $*$s, but we understand that there is a forthcoming solution [17].

Other decision procedures for fix-point logics use resolution calculi, translation methods, automata-theoretic methods, and game theoretic methods: see [1] for references. We know of no implementations for PDL based on these methods.

Here, we give a sound, complete and terminating decision procedure for PDL with the following advantages and disadvantages:

One-pass nature: our method constructs a single-rooted finite cyclic tree where all cycles connect a leaf to an ancestor. In particular, there are no cross-branch edges, meaning that our method can be used to explore the search space in a depth-first, left-to-right manner. Consequently, the space required to explore one branch can be reclaimed upon backtracking.

Pre-processing: although there is no initial construction of a potentially exponential sized automaton, nor any calls to a subroutine for deciding equality of regular languages, we do require a pre-processing stage to avoid nested occurrences of $*$ which can lead to an exponential blow-up in the size of the initial formula. The longer version of this paper will show how to avoid this step, but it complicates the rules even further.

Full proofs: our proofs of soundness and completeness are all given in full detail and use only elementary methods.

Ease of implementation: our rules are easy to implement since our tableau nodes contain sets of formulae as usual and some easily defined extra information whose manipulation requires only set intersection, set membership, and the operations of min/max on integers. Our rules incorporate these low-level details, making them cumbersome to describe.

Potential for optimisation: because of their tree structure, there is potential to optimise our tableaux using techniques which have proved successful for (one-pass) tableaux for description logics [11].

Ease of generating counter-models: the soundness proof of our tableau procedure for testing PDL-satisfiability immediately gives an effective procedure for turning an "open" tableau into a PDL-model.

Ease of generating proofs: our tableau calculus can be turned into a cut-free Gentzen-style calculus with "cyclic proofs". Unlike existing cut-free Gentzen-style calculi for fix-point logics [12, 13] we can give an optimal, rather than worst-case, bound for the finitary version of the omega rule for PDL.

Potential for parallelisation: our rules build the branches independently, but combine their results during backtracking, so it is possible to implement our procedure on a bank of parallel processors.

Working Prototype: we have implemented a (sequential) prototype of our basic tableau method using the Tableau Work Bench (`twb.rsise.anu.edu.au`) which allows users to test arbitrary PDL formulae over the web.

Double-exponential Worst Case: in the worst-case, our method requires time which is double-exponential in the size of the given formula, meaning triple exponential via the pre-processing blow-up. That is, our method may have to repeat the same work on multiple branches. Further experimental work

is required to determine if a highly optimised version of our method can be made practical for the average case.

Generality: Our method for PDL fits into a class of similar "one pass" methods for other fix-point logics like PLTL [18] and CTL [1]. The next step is to see if our methods can be turned into optimal algorithms which exhibit good average-case behaviour by using techniques like global caching which have proved useful in obtaining an easy to implement and provably correct EXPTIME algorithm for the EXPTIME-complete logic ALC [10].

2 Syntax, Semantics and Hintikka Structures

Definition 1. *Let* AFml *be a countably infinite set of propositional atoms and* APrg *a finite set of* atomic programs *with* $\mathrm{AFml} \cap \mathrm{APrg} = \emptyset$. *The set* Fml *of all formulae and the set* Prg *of all* programs *are defined inductively as follows:*

1. $\mathrm{AFml} \subseteq \mathrm{Fml}$
2. *if* $\varphi, \psi \in \mathrm{Fml}$ *then* $\neg\varphi \in \mathrm{Fml}$ *and* $\varphi \wedge \psi \in \mathrm{Fml}$ *and* $\varphi \vee \psi \in \mathrm{Fml}$
3. *if* $\varphi \in \mathrm{Fml}$ *and* $\alpha \in \mathrm{Prg}$ *then* $\langle\alpha\rangle\varphi \in \mathrm{Fml}$ *and* $[\alpha]\varphi \in \mathrm{Fml}$
4. $\mathrm{APrg} \subseteq \mathrm{Prg}$
5. *if* $\alpha \in \mathrm{Prg}$ *and* $\beta \in \mathrm{Prg}$ *then* $(\alpha;\beta) \in \mathrm{Prg}$ *and* $\alpha \cup \beta \in \mathrm{Prg}$ *and* $\alpha* \in \mathrm{Prg}$
6. *if* $\varphi \in \mathrm{Fml}$ *then* $\varphi? \in \mathrm{Prg}$.

We use p *and* q *to range over members of* AFml *and use* a *and* b *to range over members of* APrg. *A formula of the form* $\langle\alpha\rangle\varphi$ *is called a* $\langle\rangle$*-formula, and a formula of the form* $\langle\alpha*\rangle\varphi$ *is called an* eventuality. *Let* $\mathrm{Fml}\langle\rangle$ *denote the set of all* $\langle\rangle$*-formula and* $\mathrm{Fml}\langle*\rangle$ *the set of all eventualities. The notions of sub-formula and sub-program are defined as usual as are the connectives* $\rightarrow$, $\leftrightarrow$, $\top$ *and* $\bot$.

Definition 2. *A* transition frame *is a pair* (W, R) *where* W *is a non-empty set of worlds and* R *is a function that assigns to each atomic program* $a \in \mathrm{APrg}$ *a binary relation* R_a *over* W. *A* model $M = (W, R, V)$ *is a transition frame* (W, R) *and a valuation function* $V : \mathrm{AFml} \rightarrow 2^W$ *which associates with each propositional atom* p *a set* $V(p)$ *of worlds in which* p *is true.*

Definition 3. *Let* $M = (W, R, V)$ *be a model. The functions* $\tau_M : \mathrm{Fml} \rightarrow 2^W$ *and* $\rho_M : \mathrm{Prg} \rightarrow 2^{W \times W}$ *are defined inductively as follows:*

$$
\begin{array}{ll}
\tau_M(p) & := V(p) \\
\tau_M(\neg\varphi) & := W \setminus \tau_M(\varphi) \\
\tau_M(\varphi \wedge \psi) & := \tau_M(\varphi) \cap \tau_M(\psi) \\
\tau_M(\varphi \vee \psi) & := \tau_M(\varphi) \cup \tau_M(\psi) \\
\tau_M([\alpha]\varphi) & := \{w \mid \forall v \in W.\, (w,v) \in \rho_M(\alpha) \;\Rightarrow\; v \in \tau_M(\varphi)\} \\
\tau_M(\langle\alpha\rangle\varphi) & := \{w \mid \exists v \in W.\, (w,v) \in \rho_M(\alpha) \;\&\; v \in \tau_M(\varphi)\} \\
\rho_M(a) & := R_a \\
\rho_M(\alpha;\beta) & := \{(w,v) \mid \exists u \in W.\, (w,u) \in \rho_M(\alpha) \;\&\; (u,v) \in \rho_M(\beta)\} \\
\rho_M(\alpha \cup \beta) & := \rho_M(\alpha) \cup \rho_M(\beta) \\
\rho_M(\alpha*) & := \big\{(w,v) \mid \exists k \in \mathbb{N}. \exists w_0, \ldots, w_k \in W.\, \big(w_0 = w \;\&\; w_k = v \;\&\; \\
 & \qquad\qquad \forall i \in \{0, \ldots, k-1\}.\, (w_i, w_{i+1}) \in \rho_M(\alpha)\big)\big\} \\
\rho_M(\varphi?) & := \{(w,w) \mid w \in \tau_M(\varphi)\}\ .
\end{array}
$$

Table 1. Smullyan's $\boldsymbol{\alpha}$- and $\boldsymbol{\beta}$-notation to classify formulae

$\boldsymbol{\alpha}$	$\boldsymbol{\alpha}_1$	$\boldsymbol{\alpha}_2$
$\varphi \wedge \psi$	φ	ψ
$\langle \alpha; \beta \rangle \varphi$	$\langle \alpha \rangle \langle \beta \rangle \varphi$	
$[\alpha; \beta] \varphi$	$[\alpha][\beta] \varphi$	
$[\alpha \cup \beta] \varphi$	$[\alpha] \varphi$	$[\beta] \varphi$
$[\alpha *] \varphi$	φ	$[\alpha][\alpha *] \varphi$
$\langle \psi ? \rangle \varphi$	φ	ψ

$\boldsymbol{\beta}$	$\boldsymbol{\beta}_1$	$\boldsymbol{\beta}_2$
$\varphi \vee \psi$	φ	ψ
$\langle \alpha \cup \beta \rangle \varphi$	$\langle \alpha \rangle \varphi$	$\langle \beta \rangle \varphi$
$\langle \alpha * \rangle \varphi$	φ	$\langle \alpha \rangle \langle \alpha * \rangle \varphi$
$[\psi ?] \varphi$	φ	$\sim \psi$

For $w \in W$ and $\varphi \in \mathrm{Fml}$, we write $M, w \Vdash \varphi$ iff $w \in \tau_M(\varphi)$.

Definition 4. *A formula $\varphi \in \mathrm{Fml}$ is* satisfiable *iff there exists a model $M = (W, R, V)$ and some $w \in W$ such that $M, w \Vdash \varphi$. A formula $\varphi \in \mathrm{Fml}$ is* valid *iff $\neg\varphi$ is not satisfiable.*

Definition 5. *A formula $\varphi \in \mathrm{Fml}$ is in* negation normal form *if the symbol $\neg$ appears only immediately before propositional atoms. For every formula $\varphi \in \mathrm{Fml}$, we can obtain a formula $\mathrm{nnf}(\varphi)$ in negation normal form by pushing negations inward as far as possible (*e.g. *by using de Morgan's laws) such that $\varphi \leftrightarrow \mathrm{nnf}(\varphi)$ is valid. We define $\sim\varphi := \mathrm{nnf}(\neg\varphi)$.*

We use Smullyan's $\boldsymbol{\alpha}$- and $\boldsymbol{\beta}$-notation to categorise formulae as shown in Table 1. Note that we use bolding to differentiate between Smullyan's notation and the use of α and β as members of Prg. By $\boldsymbol{\alpha}$- and $\boldsymbol{\beta}$-formulae, we mean formulae which match the patterns in the first column of the two tables. By $\boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_2$, $\boldsymbol{\beta}_1$, and $\boldsymbol{\beta}_2$, we mean the corresponding formulae in the corresponding column.

Proposition 6. *In the notation of Table 1, the formulae of the form $\boldsymbol{\alpha} \leftrightarrow \boldsymbol{\alpha}_1 \wedge \boldsymbol{\alpha}_2$ and $\boldsymbol{\beta} \leftrightarrow \boldsymbol{\beta}_1 \vee \boldsymbol{\beta}_2$ are valid.*

Definition 7. *Let $\phi \in \mathrm{Fml}$ be a formula in negation normal form. The* closure $\mathrm{cl}(\phi)$ *of ϕ is the least set of formulae such that:*

1. $\phi \in \mathrm{cl}(\phi)$
2. $\langle a \rangle \varphi \in \mathrm{cl}(\phi)$ *or* $[a]\varphi \in \mathrm{cl}(\phi) \;\Rightarrow\; \varphi \in \mathrm{cl}(\phi)$
3. $\varphi \wedge \psi \in \mathrm{cl}(\phi)$ *or* $\varphi \vee \psi \in \mathrm{cl}(\phi) \;\Rightarrow\; \varphi \in \mathrm{cl}(\phi)$ & $\psi \in \mathrm{cl}(\phi)$
4. $\langle \alpha; \beta \rangle \varphi \in \mathrm{cl}(\phi) \;\Rightarrow\; \langle \alpha \rangle \langle \beta \rangle \varphi \in \mathrm{cl}(\phi)$
5. $[\alpha; \beta] \varphi \in \mathrm{cl}(\phi) \;\Rightarrow\; [\alpha][\beta] \varphi \in \mathrm{cl}(\phi)$
6. $\langle \alpha \cup \beta \rangle \varphi \in \mathrm{cl}(\phi) \;\Rightarrow\; \langle \alpha \rangle \varphi \in \mathrm{cl}(\phi)$ & $\langle \beta \rangle \varphi \in \mathrm{cl}(\phi)$
7. $[\alpha \cup \beta] \varphi \in \mathrm{cl}(\phi) \;\Rightarrow\; [\alpha] \varphi \in \mathrm{cl}(\phi)$ & $[\beta] \varphi \in \mathrm{cl}(\phi)$
8. $\langle \alpha * \rangle \varphi \in \mathrm{cl}(\phi) \;\Rightarrow\; \varphi \in \mathrm{cl}(\phi)$ & $\langle \alpha \rangle \langle \alpha * \rangle \varphi \in \mathrm{cl}(\phi)$
9. $[\alpha *] \varphi \in \mathrm{cl}(\phi) \;\Rightarrow\; \varphi \in \mathrm{cl}(\phi)$ & $[\alpha][\alpha *] \varphi \in \mathrm{cl}(\phi)$
10. $\langle \psi ? \rangle \varphi \in \mathrm{cl}(\phi) \;\Rightarrow\; \psi \in \mathrm{cl}(\phi)$ & $\varphi \in \mathrm{cl}(\phi)$
11. $[\psi ?] \varphi \in \mathrm{cl}(\phi) \;\Rightarrow \sim\psi \in \mathrm{cl}(\phi)$ & $\varphi \in \mathrm{cl}(\phi)$.

It is easy to see that all formula in $\mathrm{cl}(\phi)$ are also in negation normal form.

Definition 8. *A* structure (W, R, L) [*for* $\varphi \in \mathrm{Fml}$] *is a transition frame* (W, R) *and a labelling function* $L : W \to 2^{\mathrm{Fml}}$ *which associates with each world* $w \in W$ *a set* $L(w)$ *of formulae* [*and has* $\varphi \in L(v)$ *for some world* $v \in W$].

Definition 9. *For a given* $\varphi \in \mathrm{Fml}$ *the (infinite) set* $\mathrm{pre}(\varphi)$ *is defined as:*

$$\mathrm{pre}(\varphi) := \{\psi \in \mathrm{Fml} \mid \exists k \in \mathbb{N}.\, \exists \alpha_1, \ldots, \alpha_k \in \mathrm{Prg}.\, \psi = \langle \alpha_1 \rangle \ldots \langle \alpha_k \rangle \varphi\} \ .$$

For all formulae φ *and* ψ*, the binary relation* $\rightsquigarrow$ *on formulae is defined as:* $\varphi \rightsquigarrow \psi$ *iff (exactly) one of the following conditions is true:*

- $\exists \chi \in \mathrm{Fml}, \alpha, \beta \in \mathrm{Prg}.\, \varphi = \langle \alpha ; \beta \rangle \chi \ \&\ \psi = \langle \alpha \rangle \langle \beta \rangle \chi$
- $\exists \chi \in \mathrm{Fml}, \alpha, \beta \in \mathrm{Prg}.\, \varphi = \langle \alpha \cup \beta \rangle \chi \ \&\ \big(\psi = \langle \alpha \rangle \chi$ *or* $\psi = \langle \beta \rangle \chi\big)$
- $\exists \chi \in \mathrm{Fml}, \alpha \in \mathrm{Prg}.\, \varphi = \langle \alpha * \rangle \chi \ \&\ \big(\psi = \chi$ *or* $\psi = \langle \alpha \rangle \langle \alpha * \rangle \chi\big)$
- $\exists \chi, \phi \in \mathrm{Fml}.\, \varphi = \langle \phi ? \rangle \chi \ \&\ \psi = \chi$.

The relation $\rightsquigarrow^+$ *is the transitive closure of* $\rightsquigarrow$.

Intuitively – using the notation of Table 1 – the relation $\rightsquigarrow$ relates a $\langle\rangle$-formulae $\boldsymbol{\alpha}$ (respectively $\boldsymbol{\beta}$) that is not of the form $\langle a \rangle \chi$ to $\boldsymbol{\alpha}_1$ (respectively $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$). The notion of $\mathrm{pre}(\varphi)$ is useful because eventualities like $\langle \alpha * \rangle \varphi$ get expanded into $\langle \alpha \rangle \langle \alpha * \rangle \varphi$ and $\langle \alpha \rangle \langle \alpha * \rangle \varphi$ can then end up being decomposed into formulae of the form $\langle \alpha_1 \rangle \ldots \langle \alpha_k \rangle \langle \alpha * \rangle \varphi$. Note that $\varphi \in \mathrm{pre}(\varphi)$.

Definition 10. *Let* $H = (W, R, L)$ *be a structure,* $\varphi \in \mathrm{Fml}$ *a formula,* $\beta \in \mathrm{Prg}$ *a program, and* $w \in W$ *a state. A* fulfilling chain for (φ, β, w) in H *is a finite sequence* $(w_0, \psi_0), \ldots, (w_n, \psi_n)$ *of world-formula pairs with* $n \geq 0$ *such that:*

- $w_i \in W$, $\psi_i \in \mathrm{pre}(\varphi)$, *and* $\psi_i \in L(w_i)$ *for all* $0 \leq i \leq n$
- $w_0 = w$, $\psi_0 = \langle \beta \rangle \varphi$, $\psi_n = \varphi$, *and* $\psi_i \neq \varphi$ *for all* $0 \leq i \leq n-1$
- *for all* $0 \leq i \leq n-1$, *if* $\psi_i = \langle a \rangle \chi$ *for some* $a \in \mathrm{APrg}$ *and* $\chi \in \mathrm{Fml}$ *then* $\psi_{i+1} = \chi$ *and* $w_i \, R_a \, w_{i+1}$*; otherwise* $\psi_i \rightsquigarrow \psi_{i+1}$ *and* $w_i = w_{i+1}$.

Intuitively, each ψ_i is in the label $L(w_i)$. The sequence starts at $(w_0, \langle \beta \rangle \varphi)$, ends at (w_n, φ), and no other w_i is paired with φ. Consecutive formulae in the sequence are related by $\rightsquigarrow$ and consecutive worlds w_i and w_{i+1} are the same unless $\psi_i = \langle a \rangle \chi$, in which case $\psi_{i+1} = \chi$ and $w_i \, R_a \, w_{i+1}$ must hold. Thus the eventuality $\langle \beta \rangle \varphi \in w_0$ is fulfilled by the $\varphi \in w_n$ and w_n is β-reachable from w_0.

Definition 11. *A* pre-Hintikka structure $H = (W, R, L)$ [*for* $\varphi \in \mathrm{Fml}$] *is a structure* [*for* φ] *that satisfies the following conditions for every* $w \in W$ *where* $\boldsymbol{\alpha}$ *and* $\boldsymbol{\beta}$ *are formulae as defined in Table 1:*

$$\begin{aligned}
&\mathrm{H1} : \neg p \in L(w) \ \Rightarrow\ p \notin L(w) \\
&\mathrm{H2} : \boldsymbol{\alpha} \in L(w) \ \Rightarrow\ \boldsymbol{\alpha}_1 \in L(w) \ \&\ \boldsymbol{\alpha}_2 \in L(w) \\
&\mathrm{H3} : \boldsymbol{\beta} \in L(w) \ \Rightarrow\ \boldsymbol{\beta}_1 \in L(w) \text{ or } \boldsymbol{\beta}_2 \in L(w) \\
&\mathrm{H4} : \langle a \rangle \varphi \in L(w) \ \Rightarrow\ \exists v \in W.\, w \, R_a \, v \ \&\ \varphi \in L(v) \\
&\mathrm{H5} : [a] \varphi \in L(w) \ \Rightarrow\ \forall v \in W.\, w \, R_a \, v \ \Rightarrow\ \varphi \in L(v) \ .
\end{aligned}$$

A Hintikka structure $H = (W, R, L)$ [*for* $\varphi \in \mathrm{Fml}$] *is a pre-Hintikka structure* [*for* φ] *that additionally satisfies the following condition:*

$$\mathrm{H6} : \langle \alpha * \rangle \varphi \in L(w) \ \Rightarrow\ \textit{there exists a fulfilling chain for } (\varphi, \alpha *, w) \textit{ in } H \ .$$

Although $H3$ captures the fix-point semantics of $\langle\alpha*\rangle\varphi$ by "locally unwinding" the fix-point, it does not guarantee a *least* fix-point which requires that φ has to be true eventually. We therefore additionally need $H6$ which acts "globally" and ensures that all eventualities are fulfilled. Note that $H2$ is enough to capture the correct behaviour of $[\alpha*]\varphi$ as it has a *greatest* fix-point semantics.

Theorem 12. *A formula $\varphi \in$ Fml in negation normal form is satisfiable iff there exists a Hintikka structure for φ.*

Definition 13. *The set ϵ-free of programs which do not generate the empty word is defined inductively as:*

1. $\mathrm{APrg} \subseteq \epsilon$*-free*
2. *if $\alpha \in \epsilon$-free or $\beta \in \epsilon$-free then $\alpha;\beta \in \epsilon$-free*
3. *if $\alpha \in \epsilon$-free and $\beta \in \epsilon$-free then $\alpha \cup \beta \in \epsilon$-free.*

A program $\alpha \in$ Prg (respectively a formula $\varphi \in$ Fml) is in *-normal form *iff every sub-program $\beta*$ of α (respectively of φ) has $\beta \in \epsilon$-free.*

Proposition 14. 1. *if $\alpha \in \epsilon$-free then $\langle\alpha\rangle\psi \not\rightsquigarrow^+ \psi$ for all $\psi \in fml$*
2. *if $\alpha \in$ Prg is in *-normal form then $\langle\alpha\rangle\psi \not\rightsquigarrow^+ \langle\alpha\rangle\psi$ for all $\psi \in fml$*
3. *if $\varphi \in$ Fml is in *-normal form then $\varphi \not\rightsquigarrow^+ \varphi$*

Corollary 15. *If $\varphi \in$ Fml is in *-normal form then there exists no infinite sequence $\varphi_0 = \varphi, \varphi_1, \ldots$ such that $\varphi_i \rightsquigarrow \varphi_{i+1}$ for all $i \in \mathbb{N}$.*

Theorem 16. *For every $\varphi \in$ Fml [in negation normal form], there is a $\psi \in$ Fml in *-normal form [and in negation normal form] such that $\varphi \leftrightarrow \psi$ is valid.*

3 An Overview of the Algorithm

Our algorithm starts with a root node containing the given formula ϕ and an empty list called HCr. It builds a tree by repeatedly applying $\boldsymbol{\alpha}$- and $\boldsymbol{\beta}$-rules to decompose the formula according to the semantics of PDL. In particular, the $\boldsymbol{\beta}$-rule for $\langle\alpha^*\rangle\varphi$ has a left child which fulfils this eventuality by reducing it to φ, and a right child which procrastinates its fulfilment by "reducing" it to $\langle\alpha\rangle\langle\alpha^*\rangle\varphi$. All $\boldsymbol{\alpha}$- and $\boldsymbol{\beta}$-children inherit the HCr-value of their parents unchanged.

The application of $\boldsymbol{\alpha}$- and $\boldsymbol{\beta}$-rules stops when all leaves contain only atoms, negated atoms, and all $\langle\rangle$-formulae and all $[]$-formulae begin with outermost atomic programs only. For each such leaf node l, and for each $\langle a\rangle\xi$-formula in l, the $\langle\rangle$-rule creates a successor node containing $\xi \cup \Delta$, where $\Delta = \{\psi \mid [a]\psi \in l\}$, as is standard for modal logics, and passes the successor a new HCr-value by extending the current HCr with the pair $(\xi, \{\xi\} \cup \Delta)$ These successors are then saturated to produce new leaves using the $\boldsymbol{\alpha}$- and $\boldsymbol{\beta}$-rules, and the $\langle\rangle$-rule creates the successors of these new leaves, and so on.

If left unchecked, this procedure can produce infinite branches since the same successors can be created again and again on the same branch. To obtain termination, the $\langle\rangle$-rule creates a successor containing $\{\xi\} \cup \Delta$ for l only if the pair

$(\xi, \{\xi\} \cup \Delta) \notin \mathrm{HCr}_l$ since this indicates that this successor has not already been created previously higher up on the current branch.

If the successor $\{\xi\} \cup \Delta$ exists already, then the current branch is blocked from re-creating it. The resulting loop may be "bad" since every $\boldsymbol{\beta}$-application on this branch for an eventuality $\langle\alpha^*\rangle\varphi$ may procrastinate, so $\langle\alpha^*\rangle\varphi$ can never be fulfilled even if we were to repeat the loop ad infinitum. To flag this potentially "bad" loop, our algorithm passes up a triple $(\xi, \langle\alpha^*\rangle\varphi, h)$ to its parent if ξ is a decomposition of $\langle\alpha^*\rangle\varphi$ and h is the height of the blocking previous successor.

During backtracking, our rules modify the first component of the triples to reverse-track the decomposition of $\langle\alpha^*\rangle\varphi$ and merge "matching" triples being passed up from the children to compute the triples for the parent. In particular, a "bad" loop can be rescued to become a "good" loop at the $\boldsymbol{\beta}$-node for $\langle\alpha^*\rangle\varphi$ if the left child remains open and the right child can access the left child by following the loop down to the leaf, following the loop up to its blocking node, and returning back down to the left child. Conversely, if all such potential rescuers are closed, then some modification $(\xi', \langle\alpha^*\rangle\varphi, h)$ of the triple $(\xi, \langle\alpha^*\rangle\varphi, h)$ will survive the journey up until it reaches the blocking node at height h. The parent of this blocking node is then "closed" since there is no hope of converting the "bad" loop to a good "loop". A complication arises if both $\boldsymbol{\beta}$-children of a non-$\langle\alpha^*\rangle\varphi$-node "loop", so our algorithm keeps the triple whose blocking node is higher in the tree using a min operation since the triple then meets more potentially rescuing $\langle\alpha^*\rangle\varphi$-nodes on its way up.

If the status of the root node becomes "closed" then the formula ϕ is PDL-unsatisfiable, else the tableau contains a PDL-model for ϕ.

4 A One-pass Tableau Algorithm for *PDL*

To separate "good" and "bad" cycles, our algorithm stores additional information in each tableau node using *histories* and *variables* [18] where histories are passed from parents to children and variables are passed from children to parents.

Definition 17. *A* tableau node x *is of the form* $(\Gamma :: \mathrm{HCr} :: \mathrm{mrk}, \mathrm{uev})$ *where:*

Γ *is a set of formulae*
HCr *is a list of pairs* (φ, Δ) *where* Δ *is a set of formulae and* $\varphi \in \Delta$ *is a distinguished formula from* Δ
mrk *is a boolean valued variable indicating whether the node is marked and*
uev *is a partial function from* $\mathrm{Fml}\langle\rangle \times \mathrm{Fml}\langle*\rangle$ *to* $\mathbb{N}_{>0}$.

Definition 18. *A* tableau *for a formula set* $\Gamma \subseteq \mathrm{Fml}$ *and a list of formula sets* HCr *is a tree of tableau nodes with root* $(\Gamma :: \mathrm{HCr} :: \mathrm{mrk}, \mathrm{uev})$ *where the children of a node* x *are obtained by a single application of a rule to* x *(*i.e. *only one rule can be applied to a node). A tableau is* expanded *if no rules can be applied to any of its leaves. On any branch of a tableau, a node* t *is an* ancestor *of a node* s *iff* t *lies above* s *on the unique path from the root down to* s.

The list HCr is the only history whereas mrk and uev are variables and their values are determined by the children of the root. Informally, the value of mrk at node x is **true** if x is "closed" meaning that the set of formulae belonging to x is unsatisfiable. The history HCr at the root is empty and is used to guarantee termination as explained in Section 3. Since repeated nodes cause "loops", a node that is not "closed" is not necessarily "open" as in traditional tableaux.

Definition 19. *The partial function* $\mathrm{uev}_\perp : \mathrm{Fml}\langle\rangle \times \mathrm{Fml}\langle * \rangle \rightharpoonup \mathbb{N}_{>0}$ *is the constant function that is undefined for all pairs of formulae so that* $\mathrm{uev}_\perp(\psi_1, \psi_2) = \perp$ *for all* $\psi_1 \in \mathrm{Fml}\langle\rangle$ *and* $\psi_2 \in \mathrm{Fml}\langle * \rangle$.

The name "uev" indicates that this function tracks unfulfilled eventualities, so $\mathrm{uev}_\perp$ means that all eventualities are fulfilled, and $\mathrm{uev}(\chi_1, \chi_2)$ defined means that the eventuality χ_2 remains potentially unfulfilled. But when a node is "closed" ($\mathrm{mrk} = \mathbf{true}$), its uev becomes irrelevant and is arbitrarily set to $\mathrm{uev}_\perp$.

4.1 The Rules

To focus on the "important" parts of the rule, we use "$\cdots$" for the "unimportant" parts which are passed from node to node unchanged. We use Γ and Δ for sets of formulae, we use $[-]\Delta$ for a set of formulae all beginning with a []-modality, and write $\varphi_1\ ,\ \ldots\ ,\ \varphi_n\ ,\ \Delta_1\ ,\ \ldots\ ,\ \Delta_m$ to denote the partition $\{\varphi_1\} \uplus \cdots \uplus \{\varphi_n\} \uplus \Delta_1 \uplus \cdots \uplus \Delta_m$ of formulae in a node.

Terminal Rule.

$$(id)\ \ \frac{(\Gamma :: \cdots :: \mathrm{mrk}, \mathrm{uev})}{}\ \{p, \neg p\} \subseteq \Gamma \text{ for some } p \in \mathrm{AFml}$$

with $\mathrm{mrk} := \mathbf{true}$ and $\mathrm{uev} := \mathrm{uev}_\perp$ since it is clearly "closed".

Linear ($\boldsymbol{\alpha}$) Rules.

$$(\wedge)\ \frac{(\varphi \wedge \psi,\ \Gamma :: \cdots :: \cdots, \mathrm{uev})}{(\varphi,\ \psi,\ \Gamma :: \cdots :: \cdots, \mathrm{uev}_1)} \qquad ([\cup])\ \frac{([\alpha \cup \beta]\varphi,\ \Gamma :: \cdots :: \cdots, \mathrm{uev})}{([\alpha]\varphi,\ [\beta]\varphi,\ \Gamma :: \cdots :: \cdots, \mathrm{uev}_1)}$$

$$([;])\ \frac{([\alpha;\beta]\varphi,\ \Gamma :: \cdots :: \cdots, \mathrm{uev})}{([\alpha][\beta]\varphi,\ \Gamma :: \cdots :: \cdots, \mathrm{uev}_1)} \qquad ([*])\ \frac{([\alpha *]\varphi,\ \Gamma :: \cdots :: \cdots, \mathrm{uev})}{(\varphi,\ [\alpha][\alpha *]\varphi,\ \Gamma :: \cdots :: \cdots, \mathrm{uev}_1)}$$

$$(\langle ? \rangle)\ \frac{(\langle \psi ? \rangle \varphi,\ \Gamma :: \cdots :: \cdots, \mathrm{uev})}{(\psi,\ \varphi,\ \Gamma :: \cdots :: \cdots, \mathrm{uev}_1)} \qquad (\langle ; \rangle)\ \frac{(\langle \alpha;\beta \rangle \varphi,\ \Gamma :: \cdots :: \cdots, \mathrm{uev})}{(\langle \alpha \rangle \langle \beta \rangle \varphi,\ \Gamma :: \cdots :: \cdots, \mathrm{uev}_1)}$$

with the following conditions for the indicated rules:

$\langle ; \rangle$-rule:

$$\mathrm{uev}(\chi_1, \chi_2) := \begin{cases} \mathrm{uev}_1(\langle \alpha \rangle \langle \beta \rangle \varphi, \chi_2) & \text{if } \chi_1 = \langle \alpha;\beta \rangle \varphi \\ \mathrm{uev}_1(\chi_1, \chi_2) & \text{if } \chi_1 \in \Gamma \\ \perp & \text{otherwise} \end{cases}$$

$\langle ? \rangle$-rule:

$$\mathrm{uev}(\chi_1, \chi_2) := \begin{cases} \mathrm{uev}_2(\varphi, \chi_2) & \text{if } \chi_1 = \langle \psi ? \rangle \varphi \\ \mathrm{uev}_1(\chi_1, \chi_2) & \text{if } \chi_1 \in \Gamma \\ \bot & \text{otherwise} \end{cases}$$

all other rules:

$$\mathrm{uev}(\chi_1, \chi_2) := \begin{cases} \mathrm{uev}_1(\chi_1, \chi_2) & \text{if } \chi_1 \in \Gamma \\ \bot & \text{otherwise} \end{cases}$$

Most rules are standard except for the uev since they just capture the transformations in Table 1. The $[*]$-rule captures the fix-point nature of $[\alpha *]\varphi$ according to Prop. 6. In the case of the $\langle ; \rangle$- and the $\langle ? \rangle$-rule, the definitions of uev ensure that $\mathrm{uev}(\chi_1, \chi_2)$, where χ_1 is the principal $\langle\rangle$-formula, inherits its value from the corresponding $\langle\rangle$-formulae in the child. For example, setting $\mathrm{uev}(\langle \alpha ; \beta \rangle \varphi, \chi_2) = \mathrm{uev}_1(\langle \alpha \rangle \langle \beta \rangle \varphi, \chi_2)$ reverse-tracks the decomposition of $\langle \alpha ; \beta \rangle \varphi$ into $\langle \alpha \rangle \langle \beta \rangle \varphi$. Note that $\mathrm{uev}(\chi_1, \chi_2)$ is only defined if χ_1 is in the parent.

Universal Branching ($\boldsymbol{\beta}$) Rules.

$$(\vee) \quad \frac{(\varphi_1 \vee \varphi_2,\ \Gamma :: \cdots :: \mathrm{mrk}, \mathrm{uev})}{(\varphi_1,\ \Gamma :: \cdots :: \mathrm{mrk}_1, \mathrm{uev}_1) \mid (\varphi_2,\ \Gamma :: \cdots :: \mathrm{mrk}_2, \mathrm{uev}_2)}$$

$$(\langle \cup \rangle) \quad \frac{(\langle \alpha \cup \beta \rangle \varphi,\ \Gamma :: \cdots :: \mathrm{mrk}, \mathrm{uev})}{(\langle \alpha \rangle \varphi,\ \Gamma :: \cdots :: \mathrm{mrk}_1, \mathrm{uev}_1) \mid (\langle \beta \rangle \varphi,\ \Gamma :: \cdots :: \mathrm{mrk}_2, \mathrm{uev}_2)}$$

$$(\langle * \rangle) \quad \frac{(\langle \alpha * \rangle \varphi,\ \Gamma :: \cdots :: \mathrm{mrk}, \mathrm{uev})}{(\varphi,\ \Gamma :: \cdots :: \mathrm{mrk}_1, \mathrm{uev}_1) \mid (\langle \alpha \rangle \langle \alpha * \rangle \varphi,\ \Gamma :: \cdots :: \mathrm{mrk}_2, \mathrm{uev}_2)}$$

$$([?]) \quad \frac{([\psi ?] \varphi,\ \Gamma :: \cdots :: \mathrm{mrk}, \mathrm{uev})}{(\sim \psi,\ \Gamma :: \cdots :: \mathrm{mrk}_1, \mathrm{uev}_1) \mid (\varphi,\ \Gamma :: \cdots :: \mathrm{mrk}_2, \mathrm{uev}_2)}$$

with the following conditions for the indicated rules:

$\vee$- and $[?]$-rule: for $i = 1, 2$

$$\mathrm{uev}'_i(\chi_1, \chi_2) := \begin{cases} \mathrm{uev}_i(\chi_1, \chi_2) & \text{if } \chi_1 \in \Gamma \\ \bot & \text{otherwise} \end{cases}$$

$\langle \cup \rangle$-rule:

$$\mathrm{uev}'_1(\chi_1, \chi_2) := \begin{cases} \mathrm{uev}_1(\langle \alpha \rangle \varphi, \chi_2) & \text{if } \chi_1 = \langle \alpha \cup \beta \rangle \varphi \\ \mathrm{uev}_1(\chi_1, \chi_2) & \text{if } \chi_1 \in \Gamma \\ \bot & \text{otherwise} \end{cases}$$

$$\mathrm{uev}'_2(\chi_1, \chi_2) := \begin{cases} \mathrm{uev}_2(\langle \beta \rangle \varphi, \chi_2) & \text{if } \chi_1 = \langle \alpha \cup \beta \rangle \varphi \\ \mathrm{uev}_2(\chi_1, \chi_2) & \text{if } \chi_1 \in \Gamma \\ \bot & \text{otherwise} \end{cases}$$

$\langle * \rangle$-rule:

$$\mathrm{uev}'_1(\chi_1, \chi_2) := \begin{cases} \bot & \text{if } \chi_1 = \chi_2 = \langle \alpha * \rangle \varphi \\ \mathrm{uev}_1(\varphi, \chi_2) & \text{if } \chi_1 = \langle \alpha * \rangle \varphi \neq \chi_2 \\ \mathrm{uev}_1(\chi_1, \chi_2) & \text{if } \chi_1 \in \Gamma \\ \bot & \text{otherwise} \end{cases}$$

$$\mathrm{uev}'_2(\chi_1, \chi_2) := \begin{cases} \mathrm{uev}_2(\langle \alpha \rangle \langle \alpha * \rangle \varphi, \chi_2) & \text{if } \chi_1 = \langle \alpha * \rangle \varphi \\ \mathrm{uev}_2(\chi_1, \chi_2) & \text{if } \chi_1 \in \Gamma \\ \bot & \text{otherwise} \end{cases}$$

all rules:

$$\min_\bot(f, g)(\chi_1, \chi_2) := \begin{cases} \bot & \text{if } f(\chi_1, \chi_2) = \bot \text{ or } g(\chi_1, \chi_2) = \bot \\ \min(f(\chi_1, \chi_2), g(\chi_1, \chi_2)) & \text{otherwise} \end{cases}$$

$$\mathrm{uev} := \begin{cases} \mathrm{uev}_\bot & \text{if } \mathrm{mrk}_1 \,\&\, \mathrm{mrk}_2 \\ \mathrm{uev}'_1 & \text{if } \mathrm{mrk}_2 \,\&\, \text{not } \mathrm{mrk}_1 \\ \mathrm{uev}'_2 & \text{if } \mathrm{mrk}_1 \,\&\, \text{not } \mathrm{mrk}_2 \\ \min_\bot(\mathrm{uev}'_1, \mathrm{uev}'_2) & \text{otherwise} \end{cases}$$

$$\mathrm{mrk} := \mathrm{mrk}_1 \,\&\, \mathrm{mrk}_2$$

Most rules are standard except for the computation of uev. The $\langle * \rangle$-rule captures the fix-point nature of the eventualities, according to Prop. 6. The intuitions are:

mrk: the node is "closed" if both children are closed.

uev'_i: the definitions of uev'_i ensure that the pairs (χ_1, χ_2), where χ_1 is the principal $\langle\rangle$-formula, inherit the values from their corresponding $\langle\rangle$-formulae in the children. In the $\langle * \rangle$-rule, a special case sets the value of $\mathrm{uev}'_i(\chi_1, \chi_2)$ to $\bot$ if χ_1 and χ_2 are equal to the principal formula $\langle \alpha * \rangle \varphi$ of this rule. That is, the eventuality $\langle \alpha * \rangle \varphi$, which may be being passed up from the left branch via uev_1, is no longer unfulfilled as the left child fulfils it. Note that $\mathrm{uev}(\chi_1, \chi_2)$ is only defined if χ_1 is in the parent.

$\min_\bot$: the definition of $\min_\bot$ ensures that we take the minimum of $f(\chi_1, \chi_2)$ and $g(\chi_1, \chi_2)$ only when both functions are defined for (χ_1, χ_2).

uev: if both children are "closed" then the parent is closed via mrk, so we arbitrarily set uev as undefined. If exactly one child is closed, we take the uev$'$ of the other child. Finally, if both children are unmarked, we define uev only for all pairs of formulae that are defined both in uev'_1 and uev'_2.

The next rule sets the value of χ_2 in uev.

Existential Branching Rule.

$$(\langle\rangle) \quad \frac{\begin{array}{l} \langle a_1 \rangle \varphi_1, \ldots, \langle a_n \rangle \varphi_n,\ \langle a_{n+1} \rangle \varphi_{n+1}, \ldots, \langle a_{n+m} \rangle \varphi_{n+m},\ [-]\Delta,\ \Gamma \\ :: \mathrm{HCr} :: \mathrm{mrk}, \mathrm{uev} \end{array}}{\begin{array}{l} \varphi_1,\ \Delta_1 \\ :: \mathrm{HCr}_1 :: \mathrm{mrk}_1, \mathrm{uev}_1 \end{array} \mid \cdots \mid \begin{array}{l} \varphi_n,\ \Delta_n \\ :: \mathrm{HCr}_n :: \mathrm{mrk}_n, \mathrm{uev}_n \end{array}}$$

where:

(1) $n + m \geq 0$
(2) $\Gamma \subseteq \big(\mathrm{AFml} \cup \{\neg q \mid q \in \mathrm{AFml}\}\big)$
(3) $[-]\Delta \subseteq \big\{[a]\psi \mid a \in \mathrm{APrg}\ \&\ \psi \in \mathrm{Fml}\big\}$
(4) $\Delta_i := \{\psi \mid [a_i]\psi \in [-]\Delta\}$ for $i = 1, \ldots, n$
(5) $\forall p \in \mathrm{AFml}.\ \{p, \neg p\} \not\subseteq \Gamma$
(6) $\forall i \in \{1, \ldots, n\}.\ \forall j \in \{1, \ldots, \mathrm{len}(\mathrm{HCr})\}.\ \big(\varphi_i, \{\varphi_i\} \cup \Delta_i\big) \neq \mathrm{HCr}[j]$
(7) $\forall k \in \{n+1, \ldots, n+m\}.\ \exists j \in \{1, \ldots, \mathrm{len}(\mathrm{HCr})\}.\ \big(\varphi_k, \{\varphi_k\} \cup \Delta_k\big) = \mathrm{HCr}[j]$

with:

$$\mathrm{HCr}_i := \mathrm{HCr}\ @\ \big[\big(\varphi_i, \{\varphi_i\} \cup \Delta_i\big)\big] \text{ for } i = 1, \ldots, n$$

$$\begin{aligned}\mathrm{mrk} := {} & \textstyle\bigvee_{i=1}^{n} \mathrm{mrk}_i \text{ or} \\ & \exists i \in \{1, \ldots, n\}.\ \exists \psi \in \mathrm{Fml}\langle * \rangle. \\ & \varphi_i \in \mathrm{pre}(\psi)\ \&\ \bot \neq \mathrm{uev}_i(\varphi_i, \psi) > \mathrm{len}(\mathrm{HCr})\end{aligned}$$

$$\begin{aligned}\mathrm{uev}_k(\cdot,\cdot) := {} & j \in \{1, \ldots, len(\mathrm{HCr})\} \text{ such that } \big(\varphi_k, \{\varphi_k\} \cup \Delta_k\big) = \mathrm{HCr}[j] \\ & \text{for } k = n+1, \ldots, n+m\end{aligned}$$

$$\mathrm{uev}(\chi_1, \chi_2) := \begin{cases} \mathrm{uev}_i(\varphi_i, \chi_2) & \text{if mrk} = \textbf{false}\ \& \\ & \chi_1 = \langle a_i \rangle \varphi_i \text{ for an } i \in \{1, \ldots, n+m\}\ \& \\ & \chi_2 \in \mathrm{Fml}\langle * \rangle\ \&\ \chi_1 \in \mathrm{pre}(\chi_2) \\ \bot & \text{otherwise} \end{cases}$$

Some intuitions are in order:

(1) If $n = 0$, the application of the rule generates no new nodes and mrk vacuously evaluates to **false**. If $m = n = 0$, we additionally have uev := $\mathrm{uev}_\bot$.
(2) The set Γ contains only propositional atoms or their negations.
(3) The set $[-]\Delta$ contains only formulae of the type $[a]\varphi$. Thus (2) and (3) imply that the $\langle\rangle$-rule is applicable only if the node contains no $\boldsymbol{\alpha}$- or $\boldsymbol{\beta}$-formulae.
(4) The set Δ_i contains all formulae that must belong to the i^{th} child, which fulfils $\langle a_i \rangle \varphi_i$, so that we can build a Hintikka structure later on.
(5) The node must not contain a contradiction.
(6) If $n > 0$, then each $\langle a_i \rangle \varphi_i$ for $1 \leq i \leq n$ is not "blocked" by an ancestor and has a child containing the formula set $\varphi_i \cup \Delta_i$ thereby generating the required successor for $\langle a_i \rangle \varphi_i$. Note that len(HCr) denotes the length of HCr.
(7) If $m > 0$, then each $\langle a_k \rangle \varphi_k$ for $n + 1 \leq k \leq n + m$ is "blocked" from creating its required child $\{\varphi_k\} \cup \Delta_k$ because some ancestor does the job. This ancestor must not only consist of the formulae $\{\varphi_k\} \cup \Delta_k$ but it must also have been created to fulfil $\langle a \rangle \varphi_k$ for some $a \in \mathrm{APrg}$. Note that the values of a_k and a are not taken into account when looking for loops since we are interested only in the contents of the required child.

HCr_i: is the HCr of the parent with an extra entry to extend the "history" on the path from the root down to the i^{th} child using "@" as list concatenation. Note that we store a *pair* $(\varphi_k, \varphi_k \cup \Delta_k)$, not just $\varphi_k \cup \Delta_k$. That is, we remember that the node $\varphi_k \cup \Delta_k$ was created to fulfil $\langle a \rangle \varphi_k$ for some $a \in \mathrm{APrg}$.

mrk: captures the "existential" nature of this rule whereby the parent is marked if some child is closed or there exists a child – say the i^{th} child – and some eventuality $\langle\alpha*\rangle\chi$ "loops lower" because $\varphi_i \in \text{pre}(\langle\alpha*\rangle\chi)$ and $\text{uev}_i(\varphi_i, \langle\alpha*\rangle\chi)$ is defined and greater than the length of the current HCr. Intuitively, the latter case means that there exists an occurrence of the eventuality $\langle\alpha*\rangle\chi$ in the sub-tableau rooted at the parent which cannot be fulfilled.

uev_k: for $n+1 \leq k \leq n+m$, the k^{th} child is blocked by a proxy child higher in the branch. For every such k we set uev_k to be the *constant* function which maps every pair of formulae to the *level* j of its proxy child. Note that this is just a temporary function used to define uev as explained next. Note also that the blocking child itself must have been created to fulfil a $\langle\rangle$-formula $\langle a'\rangle\varphi_k$, as indicated by the first component of $\text{HCr}[j]$.

$\text{uev}(\chi_1, \chi_2)$: If mrk = **true** then uev is undefined everywhere. Otherwise, for each formula $\chi_1 = \langle a_i\rangle\varphi_i$ (for an $i \in \{1, \ldots, n+m\}$) and each χ_2 such that $\langle a_i\rangle\varphi_i \in \text{pre}(\chi_2)$, we take uev of $(\langle a_i\rangle\varphi_i, \chi_2)$ from the pair of formulae (φ_i, χ_2) of the corresponding (real) child if $\langle a_i\rangle\varphi_i$ is "unblocked", or set it to be the level of the proxy child higher in the branch if it is "blocked". For all other pairs of formulae, we put uev to be undefined. The intuition is that a defined $\text{uev}(\chi_1, \chi_2)$ tells us that there is a "loop" which starts at the parent and eventually "loops" up to some blocking node higher up on the current branch. The actual value of $\text{uev}(\chi_1, \chi_2)$ tells us the level of the proxy because we cannot distinguish whether this "loop" is "good" or "bad" until we backtrack up to that level. Note that the uev of a given $\langle a_i\rangle\varphi_i$ is taken from *the* child created specifically to fulfil the given φ_i, thus we can always trace the uev back down the tree if needed (as in the proofs).

The $\langle\rangle$- and *id*-rules are mutually exclusive via their side-conditions.

4.2 Termination, Soundness, and Completeness

Definition 20. *Let* $x = (\Gamma :: \text{HCr} :: \text{mrk}, \text{uev})$ *be a tableau node,* φ *a formula, and* Δ *a set of formulae. We write* $\varphi \in x$ $[\Delta \subseteq x]$ *to mean* $\varphi \in \Gamma$ $[\Delta \subseteq \Gamma]$. *The elements* HCr, mrk, *and* uev *of* x *are denoted by* HCr_x, mrk_x, *and* uev_x, *respectively. The node* x *is* marked *iff* mrk_x *is set to* **true**.

Definition 21. *Let* x *be a* $\langle\rangle$*-node in a tableau* T*, meaning that a* $\langle\rangle$*-rule was applied to* x*. Then* x *is also called a* state *and the children of* x *are called* core-nodes. *Using the notation of the* $\langle\rangle$*-rule, a formula* $\langle a_i\rangle\varphi_i \in x$ *is* blocked *iff* $n+1 \leq i \leq n+m$. *For every not blocked* $\langle a_i\rangle\varphi_i \in x$*, the* successor *of* $\langle a_i\rangle\varphi_i$ *is the* i^{th} *child of the* $\langle\rangle$*-rule. For every blocked* $\langle a_i\rangle\varphi_i \in x$ *there exists a unique core-node* y *on the path from the root of* T *to* x *such that* $\{\varphi_i\} \cup \Delta_i$ *equals the set of formulae of* y. *Moreover,* y *is the successor of a formula* $\langle a'\rangle\varphi_i$ *in the parent of* y. *We call* y *the* virtual successor *of* $\langle a_i\rangle\varphi_i$. *We also call the formula* φ_i *in the (possibly virtual) successor of* $\langle a_i\rangle\varphi_i$ *a* core-formula.

A state is another term for a $\langle\rangle$-node but a core-node can be any type of node (even a state). Core-nodes are "cores" from which states arise by $\boldsymbol{\alpha}$- and $\boldsymbol{\beta}$-rules.

Table 2. Definitions for the example in Fig. 1

UEV_i	ψ_i	χ_i
$i = 1$	$\langle a\rangle\langle a\rangle\langle(a;a)*\rangle\neg p$	$\langle(a;a)*\rangle\neg p$
$i = 2$	$\langle a;a\rangle\langle(a;a)*\rangle\neg p$	$\langle(a;a)*\rangle\neg p$
$i = 3$	$\langle(a;a)*\rangle\neg p$	$\langle(a;a)*\rangle\neg p$
$i = 4$	$\langle a\rangle\langle(a;a)*\rangle\neg p$	$\langle(a;a)*\rangle\neg p$

Proposition 22 (Termination). *Let $\phi \in$ Fml be a formula in negation normal form and in *-normal form. Any tableau T for a node $(\{\phi\} :: \cdots :: \cdots)$ is a finite tree, hence the procedure that builds a tableau always terminates.*

Let $\phi \in$ Fml be a formula in negation normal form and in *-normal form and T an expanded tableau with root $r = (\{\phi\} :: [] :: \mathrm{mrk}, \mathrm{uev})$ meaning that the initial formula set is $\{\phi\}$ and the initial HCr is the empty list with mrk and uev determined by the children of r according to the rule applied to r.

Theorem 23 (Soundness). *If the root $r \in T$ is not marked, then there exists a Hintikka structure for ϕ.*

Theorem 24 (Completeness). *For every marked node $x = (\Gamma :: \cdots :: \cdots)$ in T, the formula $\bigwedge_{\varphi\in\Gamma}\varphi$ is unsatisfiable: if r is marked, then ϕ is unsatisfiable.*

4.3 A Fully Worked Example

As an example, consider the obviously valid formula $[a*]p \to [(a;a)*]p$. Its negation $\phi := [a*]p \wedge \langle(a;a)*\rangle\neg p$ is not satisfiable and the root of an expanded tableau for ϕ should be marked. Note that ϕ is already in negation and in *-normal form.

Figure 1 shows such a tableau where the root node is node (1). Each node is classified as a ρ-node if rule ρ is applied to that node in the tableau. The unlabelled edges go from states to core-nodes. Every partial function UEV_i maps the pair of formulae (ψ_i, χ_i) as given in Table 2 to 1 and is undefined otherwise as explained below. The histories are defined as $HCR_1 := [(\varphi_1, \Delta_1)]$ where $\varphi_1 := \langle a\rangle\langle(a;a)*\rangle\neg p$ and $\Delta_1 := \{[a*]p, \langle a\rangle\langle(a;a)*\rangle\neg p\}$ and $HCR_2 := HCR_1@[(\varphi_2, \Delta_2)]$ where $\varphi_2 := \langle(a;a)*\rangle\neg p$ and $\Delta_2 := \{[a*]p, \langle(a;a)*\rangle\neg p\}$.

The marking of the nodes (1) to (3b) in Fig. 1 with **true** is straightforward. The leaf (9) is a $\langle\rangle$-node, but it is "blocked" from creating its successor containing $\Delta := \{[a*]p, \langle a\rangle\langle(a;a)*\rangle\neg p\}$ because there is a $j \in \mathbb{N}$ such that $\mathrm{HCr}_9[j] = HCR_2[j] = (\langle a\rangle\langle(a;a)*\rangle\neg p, \Delta)$: namely $j = 1$. Thus the $\langle\rangle$-rule computes $UEV_1(\langle a\rangle\varphi_1, \langle(a;a)*\rangle\neg p) = 1$ as stated above and also puts $\mathrm{mrk}_9 :=$ **false**. As node (8a) is marked, nodes (8b), (8), (7), (6), and (5) "inherit" their functions UEV_i from their unmarked children via the corresponding $\boldsymbol{\alpha}$- and $\boldsymbol{\beta}$-rules.

The crux of our procedure happens at node (4) which is a $\langle\rangle$-node with $\mathrm{HCr}_4 = []$ and hence $len(\mathrm{HCr}_4) = 0$. The $\langle\rangle$-rule therefore finds a child node (5) and a pair of formulae $(\psi, \chi) := (\langle a\rangle\langle(a;a)*\rangle\neg p, \langle(a;a)*\rangle\neg p)$ such that ψ is a

core-formula, $\psi \in \mathrm{pre}(\chi)$, and $1 = UEV(\psi,\chi) = \mathrm{uev}_5(\psi,\chi) > len(\mathrm{HCr}_4) = 0$. That is, node (4) "sees" a child (5) that "loops lower", meaning that node (5) is the root of an "isolated" subtree which does not fulfil its eventuality $\langle(a;a)*\rangle\neg p$. Thus the $\langle\rangle$-rule sets $\mathrm{mrk}_4 = \mathbf{true}$, marking (4) as "closed". The propagation of **true** to the root is then just via simple $\boldsymbol{\alpha}$- and $\boldsymbol{\beta}$-rule applications.

The closure clearly arise because both nodes (3a) and (8a) are marked by the *id*-rule. But suppose that node (8a) had not ended being marked. Then UEV_3 in node (8) would be undefined everywhere via the $\langle*\rangle$-rule, regardless of $\mathrm{uev}_{8\mathrm{b}}$. Thus $\langle(a;a)*\rangle\neg p$ in (8) can be fulfilled by following the $\boldsymbol{\beta}_1$-child (8a). Hence UEV_4 would also be undefined everywhere, and node (4) would not be marked.

References

1. P. Abate, R. Goré, and F. Widmann. One-pass tableaux for computation tree logic. In N. Dershowitz and A. Voronkov, editors, *Proc. LPAR 2007*, 2007.
2. F. Baader. Augmenting concept languages by transitive closure of roles: an alternative to terminological cycles. Technical Report, DFKI, 1990.
3. P. Balbiani. Propositional dynamic logic. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2007.
4. M. Ben-Ari, Z. Manna, and A. Pnueli. The temporal logic of branching time. In *Proceedings of Principles of Programming Languages*, 1981.
5. G. Bhat and R. Cleaveland. Efficient on-the-fly model checking for CTL*. In *Proc. LICS'95*, pages 388–397, 1995.
6. R. Cleaveland. Tableau-based model checking in the propositional mu-calculus. *Acta Informatica*, 27:725–747, 1990.
7. E. A. Emerson and J. Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *J. of Computer and System Sci.*, 30:1–24, 1985.
8. M. J. Fischer and R. E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and Systems Science*, 1979.
9. G. D. Giacomo and F. Massacci. Combining deduction and model checking into tableaux and algorithms for converse-pdl. *Inf. and Comp.* , 160:109–169, 2000.
10. R. Goré and L. A. Nguyen. Exptime tableaux for ALC using sound global caching. In *Proc. DL 2007*.
11. I. Horrocks and P. F. Patel-Schneider. Optimising description logic subsumption. *Journal of Logic and Computation*, 9(3):267–293, OUP, 1999.
12. G. Jäger and L. Alberucci. About cut elimination for logics of common knowledge. *Ann. Pure Appl. Logic*, 133(1-3):73–99, 2005.
13. G. Jäger, M. Kretz, and T. Studer. Cut-free common knowledge. *Journal of Applied Logic*, to appear.
14. D. Kozen and R. Parikh. An elementary proof of the completeness of PDL. *Theoretical Computer Science*, 14:113–118, 1981.
15. V. Pratt. Semantical considerations on Floyd-Hoare logic. In *Proceedings of the 17th IEEE Symposium on Foundations of Computer Science*, pages 109–121, 1976.
16. V. Pratt. A near-optimal method for reasoning about action. *Journal of Computer and System Sciences*, 20:231–254, 1980.
17. R. Schmidt and D. Tishkovsky. Personal communication. `http://www.cs.man.ac.uk/~schmidt/pdl-tableau/`, September 2007.
18. S. Schwendimann. A new one-pass tableau calculus for PLTL. In *Proc. TABLEAUX'98*, LNAI 1397:277-291. Springer, 1998.
19. P. Wolper. Temporal logic can be more expressive. *Inf. and Comp*, 56:72–99, 1983.

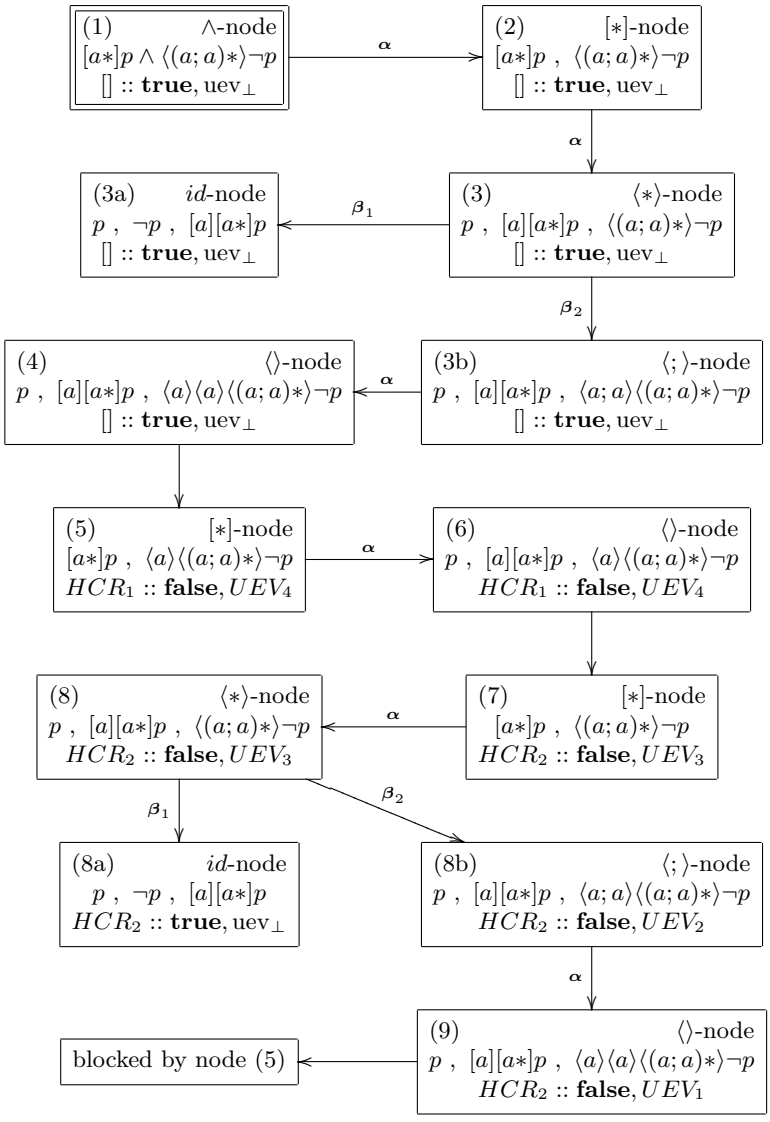

Fig. 1. An example: a closed tableau for $[a*]p \wedge \langle (a;a)* \rangle \neg p$

Appendix: Termination, Soundness and Completeness

Definition 25. *Let $G = (W, R)$ be a directed graph (*e.g. *a tableau where R is just the child-of relation between nodes). A* [full]path π *in G is a finite* [*infinite*] *sequence $x_0, x_1, x_2, \ldots$ of nodes in W such that $x_i \, R \, x_{i+1}$ for all x_i except the last node if π is finite. For $x \in W$, an* x-[full]path π *in G is a* [*full*]*path in G that has $x_0 = x$.*

Termination

Proposition 22. Let $\phi \in \mathrm{Fml}$ be a formula in negation normal form and in *-normal form. Any tableau T for a node $(\{\phi\} :: \cdots :: \cdots)$ is a finite tree, hence the procedure that builds a tableau always terminates.

Proof. It is obvious that T is a tree and that every node in T can contain only formulae of the closure $\mathrm{cl}(\phi)$. It is also not too hard to see that $\mathrm{cl}(\phi)$ is finite. Hence there are only a finite number of different sets that can be assigned to the nodes, in particular the core-nodes. As a further consequence there can only exist a finite number of pairs (φ, Δ) such that $\varphi \in \Delta \subseteq \mathrm{cl}(\phi)$. Since each core-node is assigned such a pair and the $\langle\rangle$-rule guarantees that all core-nodes on a branch possess different pairs, there can only be a finite number of core-nodes on a branch.

Furthermore, from any core-node, there are only a finite number of consecutive nodes on a branch until we reach another core-node. This claim is not obvious since eventualities and their analogues for []-formulae might "regenerate" on a branch without passing a core-node (*e.g.* $\langle a * * \rangle \varphi$). Indeed it is only true because ϕ and hence all formulae in $\mathrm{cl}(\phi)$ are in *-normal form, which rules out formulae like $\langle a * * \rangle \varphi$. The claim is basically a consequence of Coro. 15 and its analogue for []-formulae, which has not been defined in this paper.

As there are only finitely many nodes between two core-nodes and only finitely many core-nodes on any branch, all branches in T must be finite. This, the obvious fact that every node in T has finite degree, and König's lemma complete the proof sketch. □

Soundness

Theorem 23 (Soundness). If the root $r \in T$ is not marked, then there exists a Hintikka structure for ϕ.

Proof. By construction, T is a finite tree. Let T_p ("p" for pruned) be the subgraph that consists of all nodes x having the following property: there is a path of unmarked nodes from r to x inclusive. The edges of T_p are exactly the edges of T that connect two nodes in T_p. Clearly, T_p is also a finite tree with root r. Intuitively, T_p is the result of pruning all subtrees of T that have a marked root.

As *id*-nodes are marked by construction of T, all leaves of T_p must be states where all $\langle\rangle$-formulae (if any) are blocked. Hence every formula $\langle a \rangle \varphi$ of every leaf x has a virtual successor, which lies on the path from r to x. We extend T_p

to a finite cyclic tree T_l ("l" for looping) by doing the following for every leaf x: for every $\langle a\rangle\varphi \in x$, we add the edge (x, y) where y is the virtual successor of $\langle a\rangle\varphi$. These new edges are called *backward edges*.

Finally, following Ben-Ari et al. [4], the cyclic tree T_l is used to generate a structure $H = (W, R, L)$ as described next. Let W be the set of all states of T_l. For every $a \in \mathrm{APrg}$ and every $s, t \in W$, let $s\,R_a\,t$ iff s contains a formula $\langle a\rangle\psi$ and there exists a path $x_0 = s, x_1, \ldots, x_{k+1} = t$ in T_l such that x_1 is the (possibly virtual) successor of $\langle a\rangle\psi$ and each $x_i, 1 \leq i \leq k$ is an $\boldsymbol{\alpha}$- or a $\boldsymbol{\beta}$-node. Thus s is a "saturation" of x using only $\boldsymbol{\alpha}$- and $\boldsymbol{\beta}$-rules. Note that $s\,R_a\,t$ and $s\,R_b\,t$ is possible for $a \neq b$, because two formulae $\langle a\rangle\psi \in s$ and $\langle b\rangle\psi \in s$ might have the same virtual successor. It is also possible that $s\,R_a\,t$ and $s\,R_a\,u$ for $t \neq u$.

If we consider the root r of T_l as a core-node for a moment, it is not hard to see that for every state $s \in T_\mathrm{l}$ there exists a unique core-node $x \in T_\mathrm{l}$ and a unique path π of the form $x_0 = x, x_1, \ldots, x_k = s$ in T_l such that either $k = 0$ (and thus $s = x$) or $k > 0$ and each $x_i, 0 \leq i \leq k-1$ is not a state. We set $L(s)$ to be the union of all formulae of all nodes on π. Intuitively, we form $L(s)$ by adding back all the principal formulae of the $\boldsymbol{\alpha}$- and $\boldsymbol{\beta}$-rules which were applied to obtain s from x.

It is straightforward to check that H is a pre-Hintikka structure for ϕ. To show that H is even a Hintikka structure we use Lemma 26 to conclude $H6$ as is shown next.

Suppose $\langle\alpha*\rangle\varphi \in L(s)$. If we also have $\varphi \in L(s)$ then $(s, \langle\alpha*\rangle\varphi), (s, \varphi)$ is a fulfilling chain for $(\varphi, \alpha*, s)$ and we are done. Otherwise, Coro. 15 and the fact that H is a pre-Hintikka structure give us a sequence $\sigma = (s, \varphi_0), \ldots, (s, \varphi_m)$ such that:

- $\varphi_i \in \mathrm{pre}(\langle\alpha*\rangle\varphi)$ and $\varphi_i \in L(s)$ for all $0 \leq i \leq m$
- $\varphi_0 = \langle\alpha*\rangle\varphi$ and $\varphi_n = \langle a\rangle\varphi'$ for some $a \in \mathrm{APrg}$ and $\varphi' \in \mathrm{Fml}$
- $\varphi_i \rightsquigarrow \varphi_{i+1}$ for all $0 \leq i \leq m-1$.

Applying Lemma 26 for the state s and the formula $\varphi_n = \langle a\rangle\varphi'$ gives us a sequence $\sigma' := (y_0, \psi_0), \ldots, (y_n, \psi_n)$ with the properties stated in Lemma 26. Next, we replace each $y_i, 1 \leq i \leq n$ in σ' with the first state s_i that appears on the path $y_i, \ldots, y_n, \ldots, y_{n+m}$ where $y_n, \ldots, y_{n+m}$ is an arbitrary path in T_l such that y_{n+m} is a state.

It is easy to check that the combined sequence σ, σ' is a fulfilling chain for $(\varphi, \alpha*, s)$ in H if we contract all consecutive repetitions of pairs. This concludes the proof. □

Lemma 26. *Let $y \in T_\mathrm{l}$ be a node and $\psi \in y$ a formula such that $\psi \in \mathrm{pre}(\langle\alpha*\rangle\varphi)$. There exists a finite sequence $\sigma' = (y_0, \psi_0), \ldots, (y_n, \psi_n)$ of pairs with $n \geq 0$ such that:*

- *$y_0, \ldots, y_n$ is a path in T_l*
- *$y_i \in T_\mathrm{l}$, $\psi_i \in \mathrm{pre}(\varphi)$, and $\psi_i \in y_i$ for all $0 \leq i \leq n$*
- *$y_0 = y$, $\psi_0 = \psi$, $\psi_n = \varphi$, and $\psi_i \neq \varphi$ for all $0 \leq i \leq n-1$*
- *for all $0 \leq i \leq n-1$, either $\psi_i = \psi_{i+1}$ or: if $\psi_i = \langle a\rangle\chi$ for some $a \in \mathrm{APrg}$ and $\chi \in \mathrm{Fml}$ then $w_i\,R_a\,w_{i+1}$ else $\psi_i \rightsquigarrow \psi_{i+1}$.*

Proof. We inductively construct σ' starting with $(y_0, \psi_0) := (y, \psi)$. Most of the required properties of σ' follow directly from its construction and we leave it to the reader to check that they hold.

Step 1. Let (y_i, ψ_i) be the last pair of σ'. We distinguish three cases: either y_i is an $\boldsymbol{\alpha}$- or $\boldsymbol{\beta}$-node and ψ_i is not the principal formula in y_i, or y_i is an $\boldsymbol{\alpha}$- or $\boldsymbol{\beta}$-node and ψ_i is the principal formula in y_i, or y_i is a state.

If y_i is an $\boldsymbol{\alpha}$- or $\boldsymbol{\beta}$-node and ψ_i is not the principal formula in y_i, we set $\psi_{i+1} := \psi_i$ and we choose y_{i+1} to be a successor of y_i in T_l such that $\mathrm{uev}_{y_i}(\psi_i, \langle\alpha*\rangle\varphi) = \mathrm{uev}_{y_{i+1}}(\psi_{i+1}, \langle\alpha*\rangle\varphi)$. Note that such a y_{i+1} always exists since the value of $\mathrm{uev}_{y_i}(\psi_i, \langle\alpha*\rangle\varphi)$ is determined by one of its unmarked children during the construction of T and hence T_l. But it does not have to be unique. We then repeat Step 1.

If y_i is an $\boldsymbol{\alpha}$- or $\boldsymbol{\beta}$-node and ψ_i is the principal formula in y_i, we look at all pairs (x, χ) such that x is an child of y_i in T_l and ψ_i is decomposed into $\chi \in x$ and $\psi_i \rightsquigarrow \chi$ holds. By construction of T and hence T_l there is at least one unmarked child such that the corresponding pair (x, χ) obeys $\mathrm{uev}_{y_i}(\psi_i, \langle\alpha*\rangle\varphi) = \mathrm{uev}_x(\chi, \langle\alpha*\rangle\varphi)$. Let (y_{i+1}, ψ_{i+1}) be such a pair. If $\psi_{i+1} = \varphi$ we stop and return σ'; otherwise we repeat Step 1.

If y_i is a state, it is not too hard to see that ψ_i must be of the form $\langle a\rangle\chi$ for some $a \in \mathrm{APrg}$ and $\chi \in \mathrm{Fml}$. We set $(y_{i+1}, \psi_{i+1}) := (x, \chi)$ where x is the (possibly virtual) successor of $\psi_i = \langle a\rangle\chi$ and repeat Step 1. Note that if x is a non-virtual successor of ψ_i, we have $\mathrm{uev}_{y_i}(\psi_i, \langle\alpha*\rangle\varphi) = \mathrm{uev}_{y_{i+1}}(\psi_{i+1}, \langle\alpha*\rangle\varphi)$ by construction of T and hence T_l. Also note that if x is a virtual successor of ψ_i then $\psi_{i+1} = \chi$ is the core-formula of y_{i+1} by construction of T and hence T_l.

The only way for Step 1 to terminate is by finding $\psi_{i+1} = \varphi$. It is not difficult to see that the resulting (finite) sequence σ' fulfils all requirements and the proof is completed. Hence the rest of the proof shows that σ' as constructed by Step 1 is finite. Step 1 maintains the following invariant:

(†) For all appropriate $i \in \mathbb{N}$ we have $\mathrm{uev}_{y_i}(\psi_i, \langle\alpha*\rangle\varphi) = \mathrm{uev}_{y_{i+1}}(\psi_{i+1}, \langle\alpha*\rangle\varphi)$ unless y_{i+1} is the *virtual* successor of $\psi_i \in y_i$.

In other words, the values of $\mathrm{uev}_{y_i}(\psi_i, \langle\alpha*\rangle\varphi)$ and $\mathrm{uev}_{y_{i+1}}(\psi_{i+1}, \langle\alpha*\rangle\varphi)$ can differ only if (y_i, y_{i+1}) is a backward edge in T_l. We distinguish two cases: either $\mathrm{uev}_{y_0}(\psi_0, \langle\alpha*\rangle\varphi)$ is undefined or it is defined. In both cases we show that the path $y_0, y_1, \ldots$ can only have a finite number of backward edges. As every infinite path in T_l must use an infinite number of backward edges since T and T_p are finite trees, this proves that Step 1 terminates.

Case 1. If $\mathrm{uev}_{y_0}(\psi_0, \langle\alpha*\rangle\varphi)$ is undefined, the path $y_0, y_1, \ldots$ cannot contain a backward edge as shown next. Assume for a contradiction that y_i with $i \geq 0$ is the first node such that (y_i, y_{i+1}) is a backward edge. Since the initial $\mathrm{uev}_{y_0}(\psi_0, \langle\alpha*\rangle\varphi)$ was undefined, by (†) we know that $\mathrm{uev}_{y_i}(\psi_i, \langle\alpha*\rangle\varphi)$ is undefined. But y_i is a state and as $\psi_i \in y_i$, which must be of the form $\langle a\rangle\chi$ for some $a \in \mathrm{APrg}$ and $\chi \in \mathrm{Fml}$, has a virtual successor z, $\mathrm{uev}_{y_i}(\psi_i, \langle\alpha*\rangle\varphi)$ is

defined to be the height of z by the application of the $\langle\rangle$-rule to y_i during the construction of the tableau. Thus $\mathrm{uev}_{y_i}(\psi_i, \langle\alpha*\rangle\varphi)$ is both defined and undefined, which is a contradiction.

Case 2. If $h := \mathrm{uev}_{y_0}(\psi_0, \langle\alpha*\rangle\varphi)$ is defined, the path $y_0, y_1, \ldots$ can only contain a finite number of backward edges as shown next. Let y_i with $i \geq 0$ be the first node such that (y_i, y_{i+1}) is a backward edge. If no such node exists, we are obviously done. Otherwise, we have $\mathrm{uev}_{y_i}(\psi_i, \langle\alpha*\rangle\varphi) = h$ by (†). This means by construction of the tableau that there exists a set $\Delta \subseteq \mathrm{Fml}$ such that $(\psi_{i+1}, \{\psi_{i+1}\} \cup \Delta) = \mathrm{HCr}_{y_i}[h]$. Thus y_{i+1} is the h^{th} core-node on the path from the root r to y_i in T_1 and we have $\mathrm{len}(\mathrm{HCr}_{y_{i+1}}) = h$ by construction of HCr.

If $\mathrm{uev}_{y_{i+1}}(\psi_{i+1}, \langle\alpha*\rangle\varphi)$ had a value equal to or greater than h then the $\langle\rangle$-rule would cause the parent of y_{i+1} in T_1 to be marked since ψ_{i+1} is the core-formula of y_{i+1}; but we know this is not the case. Hence $\mathrm{uev}_{y_{i+1}}(\psi_{i+1}, \langle\alpha*\rangle\varphi)$ is either undefined or has a value h' that is strictly smaller than h.

If $\mathrm{uev}_{y_{i+1}}(\psi_{i+1}, \langle\alpha*\rangle\varphi)$ is undefined, we can prove exactly as in Case 1 that the path $y_{i+1}, y_{i+2}, \ldots$ cannot contain a backward edge. On the other hand, if $h' := \mathrm{uev}_{y_{i+1}}(\psi_{i+1}, \langle\alpha*\rangle\varphi)$ is defined, we can inductively repeat the arguments in Case 2 for the sequence $(y_{i+1}, \psi_{i+1}), (y_{i+2}, \psi_{i+2}), \ldots$. The induction is well-defined because of $h' < h$, meaning that eventually this inductive argument must terminate because all such h-values must be in $\mathbb{N}_{>0}$. □

Completeness

Proposition 27. *Let $M = (W, R, V)$ be a model, $w \in W$ a state and $\varphi \in \mathrm{Fml}$ a formula with $M, w \Vdash \varphi$. Furthermore, let φ be of the form $\varphi = \langle\alpha_1\rangle \ldots \langle\alpha_k\rangle\psi$ for some $\alpha_1, \ldots, \alpha_k \in \mathrm{Prg}$ and $\psi \in \mathrm{Fml}$. Then there exists a sequence $\sigma := (w_0, \psi_0), \ldots, (w_n, \psi_n)$ of world-formula pairs with $n \geq 0$ such that:*

(1) $w_i \in W$, $\psi_i \in \mathrm{pre}(\psi)$, and $M, w_i \Vdash \psi_i$ for all $0 \leq i \leq n$
(2) $w_0 = w$, $\psi_0 = \varphi$, $\psi_n = \psi$, and $\psi_i \neq \psi$ for all $0 \leq i \leq n-1$
(3) for all $0 \leq i \leq n-1$, if $\psi_i = \langle a\rangle\chi$ for some $a \in \mathrm{APrg}$ and $\chi \in \mathrm{Fml}$ then $\psi_{i+1} = \chi$ and $w_i \, R_a \, w_{i+1}$; otherwise $\psi_i \rightsquigarrow \psi_{i+1}$ and $w_i = w_{i+1}$.

Theorem 24 (Completeness). For every marked node $x = (\Gamma :: \cdots :: \cdots)$ in T, the formula $\bigwedge_{\varphi\in\Gamma} \varphi$ is not satisfiable. In particular, if r is marked, then ϕ is not satisfiable.

Proof. We use well-founded induction on the (strict) descendant relation of T. As T is a finite tree, the descendant relation is clearly well-founded. Thus we can use the following induction hypothesis for every node $x \in T$:

IH: for every descendant y of x with formula set Γ_y, if y is marked then the formula $\bigwedge_{\varphi\in\Gamma_y} \varphi$ is not satisfiable.

If a leaf $x \in T$ is marked, it must be an *id*-node as a state with no children is always unmarked. Hence, our theorem follows from the fact that $\{p, \neg p\} \subseteq x$

for some $p \in \mathrm{AFml}$. Note that this can be seen as the base case of the induction as leaves do not have descendants, hence the induction hypothesis is useless.

If x is a marked $\boldsymbol{\alpha}$-node then its child must be marked as well so we can apply the induction hypothesis and the claim follows from the fact that – in the sense of Table 1 – the formulae of the form $\boldsymbol{\alpha} \leftrightarrow \boldsymbol{\alpha}_1 \wedge \boldsymbol{\alpha}_2$ are valid (Prop. 6).

If x is a marked $\boldsymbol{\beta}$-node then both children are marked as well so we can apply the induction hypothesis and the claim follows from the fact that – in the sense of Table 1 – the formulae of the form $\boldsymbol{\beta} \leftrightarrow \boldsymbol{\beta}_1 \vee \boldsymbol{\beta}_2$ are valid (Prop. 6).

If x is a marked $\langle\rangle$-node (*i.e.* a marked state) then it has at least one child and there are two possibilities for why it was marked by the $\langle\rangle$-rule:

(1) Some child x_0 of x is marked
(2) Some unmarked child x_0 of x with core-formula φ has $\mathrm{uev}_{x_0}(\varphi, \langle\alpha*\rangle\psi) > h := \mathrm{len}(\mathrm{HCr}_x)$ for some $\alpha \in \mathrm{Prg}$ and $\psi \in \mathrm{Fml}$ with $\varphi \in \mathrm{pre}(\langle\alpha*\rangle\psi)$.

In the rest of the proof, let Γ_y denote the set of formulae of the node y. We say that a finite set of formulae Γ is satisfiable iff $\bigwedge_{\varphi\in\Gamma} \varphi$ is satisfiable.

Case 1. In the first case, it is not too hard to see that the satisfiability of Γ_x implies the satisfiability of Γ_{x_0} since the $\langle\rangle$-rule preserves satisfiability from parent to child. By the induction hypothesis, we know that Γ_{x_0} is not satisfiable, hence Γ_x cannot be satisfiable.

Case 2. In the second case, we assume that Γ_{x_0} is satisfiable and derive a contradiction. We can then prove the claim as in the first case.

So, for a contradiction, let $M = (W, R, V)$ be a model and $w \in W$ a world such that (M, w) satisfies Γ_{x_0}: that is, $M, w \Vdash \chi$ for all $\chi \in \Gamma_{x_0}$. In particular, we have $M, w \Vdash \varphi$ by assumption since $\varphi \in x_0$. As $\varphi \in \mathrm{pre}(\langle\alpha*\rangle\psi)$, it is of the form $\varphi = \langle\alpha_1\rangle \ldots \langle\alpha_k\rangle\langle\alpha*\rangle\psi$ for some $\alpha_1, \ldots, \alpha_k \in \mathrm{Prg}$. Hence Prop. 27 gives us a sequence $\sigma = (w_0, \psi_0), \ldots, (w_n, \psi_n)$ of world-formula pairs with $n \geq 0$ such that:

(a) $w_i \in W$, $\psi_i \in \mathrm{pre}(\psi)$, and $M, w_i \Vdash \psi_i$ for all $0 \leq i \leq n$
(b) $w_0 = w$, $\psi_0 = \langle\alpha_1\rangle \ldots \langle\alpha_k\rangle\langle\alpha*\rangle\psi$, $\psi_n = \psi$, and $\psi_i \neq \psi$ for all $0 \leq i \leq n-1$
(c) for all $0 \leq i \leq n-1$, if $\psi_i = \langle a\rangle\chi$ for some $a \in \mathrm{APrg}$ and $\chi \in \mathrm{Fml}$ then $\psi_{i+1} = \chi$ and $w_i \, R_a \, w_{i+1}$; otherwise $\psi_i \rightsquigarrow \psi_{i+1}$ and $w_i = w_{i+1}$.

Our plan is to show $M, w_n \nVdash \psi$, which contradicts the assumption $M, w_n \Vdash \psi_n$ since $\psi_n = \psi$. We do this by "walking down" the tableau T starting from x_0 and inductively constructing a finite sequence $\sigma' = (y_0, j_0), (y_1, j_1), \ldots$ of node-natural number pairs such that the following invariant holds where the range of i is chosen appropriately:

(1) the nodes $y_i \in T$ are unmarked and are in the subtree rooted at x_0
(2) $y_0 = x_0$, $j_0 = 0$, and either $j_{i+1} = j_i$ or $j_{i+1} = j_i + 1$
(3) each set Γ_{y_i} contains ψ_{j_i} and is satisfied by (M, w_{j_i})
(4) $\mathrm{uev}_{y_i}(\psi_{j_i}, \langle\alpha*\rangle\psi) > h$
(5) if y_i is a core-node then ψ_{j_i} is a core-formula.

Note that the singleton sequence $\sigma' = (y_0, j_0) = (x_0, 0)$ obeys the invariant. We extend σ' by repeatedly applying Step 2.

Step 2. Let (y_i, j_i) be the last pair of the sequence as constructed so far. We distinguish three cases: either y_i is an $\boldsymbol{\alpha}$- or $\boldsymbol{\beta}$-node and ψ_{j_i} is not the principal formula in y_i, or y_i is an $\boldsymbol{\alpha}$- or $\boldsymbol{\beta}$-node and ψ_{j_i} is the principal formula in y_i, or y_i is a state and ψ_{j_i}, which must then be of the form $\langle a\rangle\chi$, has a (possibly virtual) successor.

If y_i is an $\boldsymbol{\alpha}$- or $\boldsymbol{\beta}$-node and ψ_{j_i} is not the principal formula in y_i, we set $j_{i+1} := j_i$ and we choose y_{i+1} to be a successor of y_i in T such that $\Gamma_{y_{i+1}}$ is satisfied by $(M, w_{j_{i+1}}) = (M, w_{j_i})$. The existence of y_{i+1} is guaranteed by Prop. 6 and the fact that Γ_{y_i} is satisfied by (M, w_{j_i}). Note that y_{i+1} must be unmarked since otherwise the set $\Gamma_{y_{i+1}}$ would be unsatisfiable because of the induction hypothesis. Note also that we have $\mathrm{uev}_{y_{i+1}}(\psi_{j_{i+1}}, \langle\alpha*\rangle\psi) \geq \mathrm{uev}_{y_i}(\psi_{j_i}, \langle\alpha*\rangle\psi) > h$ by construction of uev.

If y_i is an $\boldsymbol{\alpha}$- or $\boldsymbol{\beta}$-node and ψ_{j_i} is the principal formula in y_i, we set $j_{i+1} := j_i+1$ and we choose y_{i+1} to be the successor of y_i that decomposes ψ_{j_i} into $\psi_{j_{i+1}}$. This is possible since we have $\psi_{j_i} \rightsquigarrow \psi_{j_{i+1}}$ by (c). We know $w_{j_i} = w_{j_{i+1}}$ by (c) and $M, w_{j_{i+1}} \Vdash \psi_{j_{i+1}}$ by (a). Moreover, the set Γ_{y_i} is satisfied by (M, w_{j_i}) by (3), and $\Gamma_{y_{i+1}} = (\Gamma_{y_i} \setminus \{\psi_{j_i}\}) \cup \{\psi_{j_{i+1}}\}$ by construction of the tableau T. Hence, the set $\Gamma_{y_{i+1}}$ is satisfied by $(M, w_{j_{i+1}})$. Note that y_{i+1} must be unmarked as otherwise the set $\Gamma_{y_{i+1}}$ would be unsatisfiable because of the induction hypothesis. Note also that we have $\mathrm{uev}_{y_{i+1}}(\psi_{j_{i+1}}, \langle\alpha*\rangle\psi) \geq \mathrm{uev}_{y_i}(\psi_{j_i}, \langle\alpha*\rangle\psi) > h$ by construction of uev.

If y_i is a state and ψ_{j_i}, which must then be of the form $\langle a\rangle\chi$ for some $a \in \mathrm{APrg}$ and $\chi \in \mathrm{Fml}$, has a (possibly virtual) successor z containing $\psi_{j_{i+1}} = \chi$ then we set $j_{i+1} := j_i + 1$ and $y_{i+1} := z$. Note that $\psi_{j_{i+1}}$ is the core-formula of the core-state y_{i+1}. If y_{i+1} is a virtual successor, a glance at the definition of uev_{y_i} in the $\langle\rangle$-rule reveals that y_{i+1} must lie on the path from x_0 to y_i (it could be x_0) as we have $\mathrm{uev}_{y_i}(\psi_{j_i}, \langle\alpha*\rangle\psi) > h$ and $h = \mathrm{len}(\mathrm{HCr}_x)$. Hence, there is a $k \leq i$ such that $y_k = y_{i+1}$. Moreover, we have $\psi_{j_k} = \psi_{j_{i+1}}$ by (5) as the core-state $y_k = y_{i+1}$ has only one core-formula. Thus, y_{i+1} is unmarked and has $\mathrm{uev}_{y_{i+1}}(\psi_{j_{i+1}}, \langle\alpha*\rangle\psi) > h$ as we have already established these facts on our way from x_0 down to y_i. If y_{i+1} is a child of y_i (*i.e.* not a virtual successor), it follows directly from the facts about y_i and the definition of the $\langle\rangle$-rule that y_{i+1} is unmarked and has $\mathrm{uev}_{y_{i+1}}(\psi_{j_{i+1}}, \langle\alpha*\rangle\psi) > h$.

Next we deduce that $(M, w_{j_{i+1}})$ satisfies $\Gamma_{y_{i+1}}$. By definition of the $\langle\rangle$-rule, $\Gamma_{y_{i+1}}$ is of the form $\psi_{j_{i+1}} \cup \Delta$ where $[a]\Delta \subseteq \Gamma_{y_i}$. We know $M, w_{j_{i+1}} \Vdash \psi_{j_{i+1}}$ by (a). We also know that (M, w_{j_i}) in particular satisfies $[a]\Delta$ since we have established that $\Gamma_{y_i} \supseteq [a]\Delta$ is satisfied by (M, w_{j_i}). As $w_{j_{i+1}}$ is a successor world of w_{j_i} (*i.e.* $w_{j_{i+1}} R_a w_{j_i}$) by (c), this implies that $(M, w_{j_{i+1}})$ satisfies Δ, and hence $\Gamma_{y_{i+1}}$.

We repeat Step 2 until we extend σ' by the first pair (y_{i+1}, j_{i+1}) such that $j_{i+1} = n$. It is an easy consequence from the construction of σ' that we construct such a pair after a finite number of steps. Then we have $\psi_{j_{i+1}} = \psi$ and it is not too hard to see that $\psi_i = \langle\alpha*\rangle\psi$ and of course $j_i = n - 1$. As we

have $\mathrm{uev}_{y_i}(\psi_{j_i}, \langle\alpha*\rangle\psi) > h$ by (4), the first child z_1 of y_i must be marked; otherwise, $\mathrm{uev}_{y_i}(\psi_{j_{i+1}}, \langle\alpha*\rangle\psi) = \mathrm{uev}_{y_i}(\langle\alpha*\rangle\psi, \langle\alpha*\rangle\psi)$ would be undefined by the $\langle*\rangle$-rule. By the induction hypothesis, Γ_{z_1} is unsatisfiable. But as $\Gamma_{z_1} \setminus \{\psi\}$ is a subset of Γ_{y_i}, which is satisfied by (M, w_{j_i}), this entails $M, w_{j_i} \nVdash \psi$. As we have $w_{j_i} = w_{j_{i+1}}$ by definition of σ, this gives us $M, w_n \nVdash \psi$, which contradicts our assumption that $M, w_n \Vdash \psi_n$. □