Invariant-Free Deduction for CTL*: The Tableau Method [§]

Technical Report. Updated on March 8th, 2010

Alexander Bolotov[1], Jose Gaintzarain[2], and Paqui Lucio[3]

[1] University of Westminster, HA1 3TP-London, UK.
[2] The University of the Basque Country, 48012-Bilbao, Spain.
[3] The University of the Basque Country, 20080-San Sebastián, Spain.

Abstract. We define a classical style, one-pass tableau for the full branching-time logic CTL*. This work extends previously defined tableau technique for propositional linear-time temporal logic, PLTL, giving a new decision procedure for CTL*. One of the core features of this method is that unlike any other known deduction for the full branching-time logic, it does not require any additional structures to deal with eventualities. Consequently the presented tableau method opens prospect for defining a dual sequent calculus which is cut-free and, in particular, invariant-free. We also hope that this tableau method could serve as a first-step towards an invariant-free resolution method for CTL*.

Keywords: Temporal Logic, Branching-time Temporal Logic, Full Computation Tree Logic, One-pass Tableaux, Invariant-Free Temporal Deduction.

1 Introduction

Temporal logic is important in many areas of computer science providing an ideal tool for specifying and modeling the behavior of various classes of dynamic systems, such as reactive systems, digital circuits, concurrent programs, etc. When there is a need for a model for a non-deterministic behaviour, with many possible futures, branching-time logics become essential. Here, most of the research has concentrated on the development of the specification framework given by the Computation Tree Logic (CTL) [3] and a number of its extensions. These extensions are due to the various syntactic restrictions and they culminate in so called Full Computation Tree Logic, CTL* [4] which does not have any restrictions on the way how the temporal formulae are assembled.

Developing deductive techniques for branching-time logics means at the same time defining the verification techniques for the corresponding specifications. Referring an interested reader to [11] and specifically, to [8], for the survey of tableau methods, the subject of this paper, here, we only note that the traditional tableau methods for temporal

[§] This work has been partially supported by the Spanish Project TIN2007-66523 and the Basque Project LoRea GIU07/35.

logic, like the one used in [4], generate auxiliary graphs for some given input which are subsequently checked and possibly pruned. In this paper, we introduce a tableau system for the Full Computation Tree Logic ($\mathsf{CTL}^\star$) in a different way. We provide a Branching Tableau Method (BTM) which does not require auxiliary graphs to decide if a set of well-formed $\mathsf{CTL}^\star$-formulae is satisfiable. We define a one-pass tableau system, avoiding the construction of auxiliary graph and this second, pruning, phase, and thus giving a new decision procedure for $\mathsf{CTL}^\star$. The only tableau decision procedure for $\mathsf{CTL}^\star$ is due to Reynolds [10]. Our system, is on the one hand, more classical, and, on the other hand, it incorporates a method that limits the length of branches in an efficient way, a problem that stayed open in [10].

The paper is organized as follows: §2 reviews the syntax and semantics of the logic $\mathsf{CTL}^\star$; §3 introduces main concepts used in the tableau construction, and describes the set of tableau rules; §4 presents an algorithm that systematically constructs a tableau for any finite set of $\mathsf{CTL}^\star$ formulae, shows that the tableau construction terminates for any finite set of formulae and gives two examples of the tableau construction; §5 provides the correctness argument, establishing the soundness and the completeness, and, finally, §6, draws concluding remarks and identifies future work.

2 Syntax, Semantics and Model Theory

The formulae of $\mathsf{CTL}^\star$ are built using the constant propositions $\mathbf{F}$ and $\mathbf{T}$, propositional variables (denoted by lowercase letters $p, q, \ldots$) from a set Prop, the classical connectives $\neg$, $\wedge$ and $\vee$, the temporal connectives "next" – $\circ$, "until" – $\mathcal{U}$, "eventually" – $\diamond$ and "always" – $\Box$ and the path quantifiers "in every path" – A and "in some path" – E. The class of *well-formed* $\mathsf{CTL}^\star$*-formulae* is recursively defined as the set of *state formulae* by $stt_1 - stt_3$:

(stt_1) $\mathbf{F}$, $\mathbf{T}$ and any propositional symbol from Prop are state formulae
(stt_2) if φ and ψ are state formulae, then so are $\neg\varphi$, $\varphi \wedge \psi$ and $\varphi \vee \psi$
(stt_3) if φ is a *path formula* (pth, see below), then $\mathsf{A}\varphi$ and $\mathsf{E}\varphi$ are state formulae
(pth_1) if φ is a state formula, then φ is also a path formula
(pth_2) if φ and ψ are path formulae, then so are $\neg\varphi$, $\varphi \wedge \psi$ and $\varphi \vee \psi$
(pth_3) if φ and ψ are path formulae, then so are $\circ\varphi$, $\varphi\,\mathcal{U}\,\psi$, $\diamond\varphi$ and $\Box\varphi$.

Some notation. When it is not important whether we speak about state or path formulae we use the term *formula* without further specification. We use the notion of *non-state formula* to refer to path formulae that are not state formulae.

In the text we will also agree that:

- Lowercase Greek letters ($\varphi, \psi, \chi, \gamma, \ldots$) denote state or path formulae.
- Uppercase Greek letters ($\Phi, \Delta, \Gamma, \Psi, \Omega, \ldots$) denote finite sets of state or path formulae.[4]
- Formulae of the form ε and $\neg\varepsilon$, where $\varepsilon \in \mathsf{Prop}$, are called *literals*.
- Q abbreviates any of the two path quantifiers.
- $\mathsf{Q}^b\varphi$ denotes φ if $b = 0$ and $\mathsf{Q}\varphi$ if $b = 1$.
- Expressions of the form $\varphi\,\mathcal{U}\,\psi$, $\diamond\varphi$ and $\neg\Box\varphi$ are called *eventualities*.

[4] Since $\mathsf{CTL}^\star$ is a non-compact logic, in this paper we only deal with finite sets of formulae.

– Eventualities of the form $\varphi \mathcal{U} \psi$ are also called *until formulae*.
– Formulae of the form $\mathbf{F}$, $\mathbf{T}$, $\neg\mathbf{F}$, $\neg\mathbf{T}$, ε, $\neg\varepsilon$ (with $\varepsilon \in \mathsf{Prop}$) and $\mathsf{Q}^b \circ \varphi$ are called *elementary*. Sets of elementary formulae are also called elementary.

Below we recall the semantics of $\mathsf{CTL}^\star$ where we evaluate formulae in tree structures where branches are ω-long sequences of states. We start with the definition of a *pre-structure* (Definition 1), introduce more concepts related to tree structures, then Definition 3 gives the conditions for a pre-structure to become a *structure*.

Definition 1 ($\mathsf{CTL}^\star$**-pre-structure).** *A $\mathsf{CTL}^\star$-pre-structure $\mathcal{M}$ is a triple of the form $(S_\mathcal{M}, R_\mathcal{M}, V_\mathcal{M})$ such that $S_\mathcal{M}$ is a set of states, $R_\mathcal{M} \subset S_\mathcal{M} \times S_\mathcal{M}$ is a binary relation over $S_\mathcal{M}$ and $V_\mathcal{M}$ is a mapping $V_\mathcal{M} : S_\mathcal{M} \to 2^{\mathsf{Prop}}$ such that*

(a) *There is a designated state in $S_\mathcal{M}$ that is called the* root *of $\mathcal{M}$ –and denoted as $\mathsf{root}(\mathcal{M})$– such that the set $\{s \in S_\mathcal{M} \mid (s, \mathsf{root}(\mathcal{M})) \in R_\mathcal{M}\}$ is empty.*
(b) *The set $\{s' \in S_\mathcal{M} \mid (s, s') \in R_\mathcal{M}\}$ is non-empty for every $s \in S_\mathcal{M}$.*
(c) *For every $s \in S_\mathcal{M}$ there exists a sequence of states $s_0, s_1, \ldots, s_n$ with $n \geq 0$ such that s_0 is the root, $s = s_n$ and $(s_i, s_{i+1}) \in R_\mathcal{M}$ for every $i \in \{0, \ldots, n-1\}$.* ∎

A *path* in $\mathcal{M}$ is a sequence of states $s_0, s_1, \ldots$ such that $(s_i, s_{i+1}) \in R_\mathcal{M}$, for every $i \geq 0$. The first state, s_0, of a path $\pi = s_0, s_1, \ldots$ is denoted $\mathsf{first}(\pi)$. A path π is a *fullpath* if π is infinite and $\mathsf{first}(\pi)$ is the root of the considered $\mathcal{M}$. We denote by $\mathsf{paths}(\mathcal{M})$ the set of non-empty paths of $\mathcal{M}$ and by $\mathsf{fullpaths}(\mathcal{M})$ the set of all fullpaths of $\mathcal{M}$. Given a path $\pi = s_0, s_1, \ldots, s_i, \ldots$ with $i \geq 1$, the path $\pi' = s_i, s_{i+1}, \ldots$ is a *suffix* of π and we denote it as $\mathsf{suffix}(\pi, s_i)$. Given paths $\pi = s_0, s_1, \ldots, s_i$ and $\pi' = s'_0, s'_1, \ldots$ such that $s'_0 = s_i$, a path $\lambda = s_0, s_1, \ldots, s_{i-1} \cdot s'_0, s'_1 \ldots$ is a path that is a concatenation of π and π'.

The most standard $\mathsf{CTL}^\star$-semantics imposes three requirements on paths.

Definition 2 (suffix, fusion and limit closures). *Given a pre-structure $\mathcal{M}$, the set $paths(\mathcal{M})$ is*

– suffix closed *if, for every path, $\pi \in \mathsf{paths}(\mathcal{M})$, ($\pi = s_0, s_1, s_2, \ldots$) for every j, $(0 \leq j)$, $\mathsf{suffix}(\pi, s_j) \in \mathsf{paths}(\mathcal{M})$.*
– fusion closed *if for every paths $\pi, \pi' \in \mathsf{paths}(\mathcal{M})$, (where $\pi = s_0, s_1, \ldots, s_i, \ldots$ and $\pi' = s'_0, s'_1, \ldots$) such that $s'_0 = s_i$, the path $\pi'' = s_0, s_1, \ldots, s_{i-1} \cdot \pi'$ is also in* $\mathsf{paths}(\mathcal{M})$,
and
– limit closed *if whenever every path in the infinite collection*[5] $\{\sigma_0 \cdot \pi_0, \ \ \sigma_0 \cdot \sigma_1 \cdot \pi_1, \ \sigma_0 \cdot \sigma_1 \cdot \sigma_2 \cdot \pi_2, \ldots\}$ *is in* $\mathsf{paths}(\mathcal{M})$ *then so is $\sigma_0 \cdot \sigma_1 \cdot \sigma_2 \cdot \sigma_3 \cdot \sigma_4 \cdot \sigma_5 \cdot \ldots$.* ∎

Limit closure requires $\mathsf{paths}(\mathcal{M})$ to contain the limit of any infinite sequence of prefixes of paths in $\mathsf{paths}(\mathcal{M})$.

Definition 3 ($\mathsf{CTL}^\star$ **Structure).** *Let $\mathcal{M}$ be a $\mathsf{CTL}^\star$-pre-structure such that* $\mathsf{paths}(\mathcal{M})$ *is suffix closed and* $\mathsf{fullpaths}(\mathcal{M})$ *is fusion and limit closed. Then $\mathcal{M}$ is a* $\mathsf{CTL}^\star$-structure. ∎

[5] where each σ_i is finite.

Note that every state in a $\mathsf{CTL}^\star$-structure $\mathcal{M}$ has a successor (seriality) and belongs to some full-path (totality). Additionaly, every infinite path in a $\mathsf{CTL}^\star$-structure is isomorphic to the natural numbers[6].

The truth of state and path formulae in a $\mathsf{CTL}^\star$-structure $\mathcal{M}$ is inductively defined below where $\langle \mathcal{M}, s\rangle \models \varphi$ denotes the truthfulness of a state formula φ in a state s of a $\mathcal{M}$ and $\langle \mathcal{M}, \pi\rangle \models \varphi$ denotes the truthfulness of a path formula φ along a path $\pi = s_0, s_1, \ldots$ of $\mathcal{M}$.

- $\langle \mathcal{M}, s\rangle \models \mathbf{T}$
- $\langle \mathcal{M}, s\rangle \not\models \mathbf{F}$
- $\langle \mathcal{M}, s\rangle \models \varepsilon$ iff $\varepsilon \in V_{\mathcal{M}}(s)$ for $\varepsilon \in \mathsf{Prop}$
- $\langle \mathcal{M}, s\rangle \models \neg\varphi$ iff $\langle \mathcal{M}, s\rangle \not\models \varphi$
- $\langle \mathcal{M}, s\rangle \models \varphi \wedge \psi$ iff $\langle \mathcal{M}, s\rangle \models \varphi$ and $\langle \mathcal{M}, s\rangle \models \psi$
- $\langle \mathcal{M}, s\rangle \models \varphi \vee \psi$ iff $\langle \mathcal{M}, s\rangle \models \varphi$ or $\langle \mathcal{M}, s\rangle \models \psi$
- $\langle \mathcal{M}, s\rangle \models \mathsf{A}\varphi$ iff $\langle \mathcal{M}, \pi\rangle \models \varphi$ for every infinite path $\pi \in \mathsf{paths}(\mathcal{M})$ such that $\mathsf{first}(\pi) = s$
- $\langle \mathcal{M}, s\rangle \models \mathsf{E}\varphi$ iff $\langle \mathcal{M}, \pi\rangle \models \varphi$ for some infinite path $\pi \in \mathsf{paths}(\mathcal{M})$ such that $\mathsf{first}(\pi) = s$
- $\langle \mathcal{M}, \pi\rangle \models \varphi$ iff $\langle \mathcal{M}, \mathsf{first}(\pi)\rangle \models \varphi$, for a state formula φ
- $\langle \mathcal{M}, \pi\rangle \models \neg\varphi$ iff $\langle \mathcal{M}, \pi\rangle \not\models \varphi$
- $\langle \mathcal{M}, \pi\rangle \models \varphi \wedge \psi$ iff $\langle \mathcal{M}, \pi\rangle \models \varphi$ and $\langle \mathcal{M}, \pi\rangle \models \psi$
- $\langle \mathcal{M}, \pi\rangle \models \varphi \vee \psi$ iff $\langle \mathcal{M}, \pi\rangle \models \varphi$ or $\langle \mathcal{M}, \pi\rangle \models \psi$
- $\langle \mathcal{M}, \pi\rangle \models \circ\varphi$ iff $\langle \mathcal{M}, \mathsf{suffix}(\pi, s_1)\rangle \models \varphi$
- $\langle \mathcal{M}, \pi\rangle \models \varphi \mathcal{U} \psi$ iff there exists $k \geq 0$ such that $\langle \mathcal{M}, \mathsf{suffix}(\pi, s_k)\rangle \models \psi$ and for every $j \in \{0, \ldots, k-1\}$ it holds $\langle \mathcal{M}, \mathsf{suffix}(\pi, s_j)\rangle \models \varphi$
- $\langle \mathcal{M}, \pi\rangle \models \Diamond\varphi$ iff there exists $k \geq 0$ such that $\langle \mathcal{M}, \mathsf{suffix}(\pi, s_k)\rangle \models \varphi$
- $\langle \mathcal{M}, \pi\rangle \models \Box\varphi$ iff $\langle \mathcal{M}, \mathsf{suffix}(\pi, s_k)\rangle \models \varphi$ for every $k \geq 0$

The semantics is extended from formulae to sets of formulae straightforwardly, i.e. given a set of state formulae Φ, $\langle \mathcal{M}, s\rangle \models \Phi$ iff $\langle \mathcal{M}, s\rangle \models \gamma$ for all $\gamma \in \Phi$, and given a set of formulae Ψ, $\langle \mathcal{M}, \pi\rangle \models \Psi$ iff $\langle \mathcal{M}, \pi\rangle \models \gamma$ for all $\gamma \in \Psi$.

For technical convenience, we extend the notion of a model to (arbitrary) sets of formulae and we say that $\mathcal{M}$ is a *model* of a set of formulae Ψ, in symbols $\mathcal{M} \models \Psi$, iff $\langle \mathcal{M}, \pi\rangle \models \Psi$ for some full-path π of $\mathcal{M}$. A satisfiable set of formulae has at least one model, otherwise it is unsatisfiable. When every structure $\mathcal{M}$ is a model of a set of formulae Ψ then Ψ is *valid*.

In order to construct models for satisfiable sets of formulae we use *cyclic* $\mathsf{CTL}^\star$-structures that we define in terms of paths over cycling sequences.

Any infinite sequence $e_0, e_1, \ldots, e_k, \ldots$ involves an implicit successor relation, namely R, such that $(e_i, e_{i+1}) \in R$ for all $i \in I\!N$. When convenient, we will write nRn' to denote $(n, n') \in R$. A finite sequence gives also a corresponding implicit successor relation with a pair for each element except for the last one. A finite sequence $S = e_0, e_1, \ldots, e_k$ is said to be *cyclic* iff its successor relation extends the implicit R with a pair (e_k, e_j) for some $0 \leq j \leq k$ (see Figure 1). Then, $e_j, \ldots, e_k$ is called

[6] We may assume, without loss of generality, that every $\mathsf{CTL}^\star$-structure is a tree ([5])

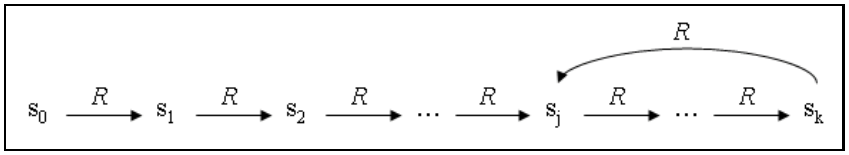


Fig. 1. Cyclic sequence of states

the *loop* of S, e_j is called the *cycling element* of S, and the *path* over S is the infinite sequence

$$\mathsf{path}(S) = e_0, e_1, \ldots, e_{j-1} \cdot \langle e_j, e_{j+1}, \ldots, e_k \rangle^\omega$$

where $\_ \cdot \_$ is the infix operator of concatenation of sequences and U^ω denotes the infinite sequence that results by concatenation of the sequence U infinitely many times. Naturally, for any non-cyclic finite sequence S we consider that $\mathsf{path}(S) = S$.

Definition 4. *A* CTL**-structure* $\mathcal{M}$ *is* cyclic *if every full-path of* $\mathcal{M}$ *is a path over a cyclic sequence of states.* ■

α-rule	α	$A(\alpha)$
$(\mathsf{A}^b\wedge)$	$\mathsf{A}^b(\varphi \wedge \psi)$	$\{\mathsf{A}^b\varphi, \mathsf{A}^b\psi\}$
$(\mathsf{E}\wedge)$	$\mathsf{E}(\gamma \wedge \psi)$	$\{\gamma, \mathsf{E}\psi\}$
(Qdel)	$\mathsf{Q}\gamma$	$\{\gamma\}$
$(\neg\mathsf{A})$	$\neg\mathsf{A}\varphi$	$\{\mathsf{A}\neg\varphi\}$
$(\neg\mathsf{E})$	$\neg\mathsf{E}\varphi$	$\{\mathsf{A}\neg\varphi\}$

β-rule	β	$B_1(\beta)$	$B_2(\beta)$
$(\mathsf{A}\vee)$	$\mathsf{A}(\gamma \vee \psi)$	$\{\gamma\}$	$\{\mathsf{A}\psi\}$
$(\mathsf{E}^b\vee)$	$\mathsf{E}^b(\varphi \vee \psi)$	$\{\mathsf{E}^b\varphi\}$	$\{\mathsf{E}^b\psi\}$

ϱ-rule	ϱ	$R(\varrho)$
$(\neg\neg)$	$\neg\neg\varphi$	φ
$(\neg\vee)$	$\neg(\varphi \vee \psi)$	$\neg\varphi \wedge \neg\psi$
$(\neg\wedge)$	$\neg(\varphi \wedge \psi)$	$\neg\varphi \vee \neg\psi$
$(\wedge\vee)$	$(\varphi \wedge \psi) \vee \chi$	$(\varphi \vee \chi) \wedge (\psi \vee \chi)$
$(\vee\wedge)$	$(\varphi \vee \psi) \wedge \chi$	$(\varphi \wedge \chi) \vee (\psi \wedge \chi)$
$(\mathcal{U})_1$	$(\varphi \mathcal{U} \psi)$	$\psi \vee (\varphi \wedge \circ(\varphi \mathcal{U} \psi))$
$(\neg\mathcal{U})$	$\neg(\varphi \mathcal{U} \psi)$	$\neg\psi \wedge (\neg\varphi \vee \neg\circ(\varphi \mathcal{U} \psi))$

ϱ-rule	ϱ	$R(\varrho)$
$(\neg\circ)$	$\neg\circ\varphi$	$\circ\neg\varphi$
$(\wedge\circ)$	$\circ\varphi \wedge \circ\psi$	$\circ(\varphi \wedge \psi)$
$(\vee\circ)$	$\circ\varphi \vee \circ\psi$	$\circ(\varphi \vee \psi)$
$(\Box)$	$\Box\varphi$	$(\varphi \wedge \circ\Box\varphi)$
$(\Diamond)_1$	$\Diamond\varphi$	$(\varphi \vee \circ\Diamond\varphi)$
$(\neg\Box)_1$	$\neg\Box\varphi$	$(\neg\varphi \vee \neg\circ\Box\varphi)$
$(\neg\Diamond)$	$\neg\Diamond\varphi$	$(\neg\varphi \wedge \neg\circ\Box\varphi)$

ϱ-cd-rule	ϱ	$R(\varrho, \Delta)$
$(\mathcal{U})_2$	$\varphi \mathcal{U} \psi$	$\psi \vee (\varphi \wedge \circ((\varphi \wedge \neg\Delta) \mathcal{U} (\psi \wedge \neg\Delta)))$
$(\Diamond)_2$	$\Diamond\varphi$	$\varphi \vee \circ((\neg\Delta) \mathcal{U} (\varphi \wedge \neg\Delta))$
$(\neg\Box)_2$	$\neg\Box\varphi$	$\neg\varphi \vee \circ((\neg\Delta) \mathcal{U} ((\neg\varphi) \wedge \neg\Delta))$

Fig. 2. The Tableau System BTM (where φ, ψ, χ represent any formula, γ represents a state formula and Δ is the context in the node and the order of the subformulae is not relevant).

3 The Tableau System

In this section we present a system of tableau rules for $\mathsf{CTL}^\star$, which we call BTM. Our tableau system is one-pass in the sense that it does not require a second phase to prune branches that contain unfulfillable eventualities. A tableau is a tree-like structure where each node n is labelled with a set of formulae[7] $L(n)$. The root is labeled with the set whose satisfiability we wish to check. The children of a node n are obtained by applying one of the rules to one of the formulae in $L(n)$. Nodes are organized in branches, so that the rules serve to either enlarge the branch (with one new child) or split the branch with two or more new children. A branch is closed if it contains some obvious inconsistency. Classically, a tableau is closed if all its branches are closed. In our case, we have a kind of nodes, called key nodes, that are inconsistent whenever at least one of the branches (of a bundle of branches, see below) starting on them is closed.

A tableau rule is applied to a set of formulae $L(n)$ labeling a node n that is the last node of a branch. Each rule application requires the selection of a designated formula from $L(n)$. Formulae are primarily classified as α-formulae and β-formulae. By means of the α- and β-rules in Figure 2, any α-formula is decomposed in a unique set, called $A(\alpha)$, and any β-formula is decomposed into two constituent sets $B_1(\beta)$ and $B_2(\beta)$. If the designated formula in $L(n)$ (for n being the last node of some branch) is an α-formula, then we enlarge the branch with a node with label $(L(n) \setminus \{\alpha\}) \cup A(\alpha)$. If the designated formula in $L(n)$ is a β-formula, then we split the branch with two nodes with respective labels $(L(n) \setminus \{\beta\}) \cup B_1(\beta)$ and $(L(n) \setminus \{\beta\}) \cup B_2(\beta)$.

Since the equivalences $\mathsf{E}(\varphi \wedge \psi) \equiv \mathsf{E}\varphi \wedge \mathsf{E}\psi$ and $\mathsf{A}(\varphi \vee \psi) \equiv \mathsf{A}\varphi \vee \mathsf{A}\psi$ only hold when φ or ψ (at least one of them) is a state formula, a complete decomposition of $\mathsf{CTL}^\star$-formulae is not possible in general. Indeed, our system BTM includes the so-called ϱ-rules that affect subformulae of the designated formula in the last node $L(n)$. The affected subformulas are called ρ-formulas and their forms are fixed by the corresponding tables in Fig. 2. Here the aim is to decompose the disjunction (resp. conjunction) inside a quantifier A (resp. E) until we obtain formulae of the form $\mathsf{A}(\bigvee_{i=1}^{n} \circ\psi_i)$ (resp. $\mathsf{E}(\bigwedge_{i=1}^{n} \circ\psi_i)$). In order to denote that ψ is a subformula inside φ we write $\varphi[\psi]$. Given a formula $\varphi[\psi] \in L(n)$ such that ψ fits some ϱ-formula (see Fig. 2), the corresponding ϱ-rule replaces $\varphi[\psi]$ by the formula that is obtained by substituting the subformula ψ by $R(\psi)$ in φ.

The system BTM also includes a particular kind of ϱ-rules, called ϱ-context-dependent-rules (shortly ϱ-cd-rules, see Fig. 2), in which the constituent also depends on the so-called *context*. The following are two remarkable issues about the application of the ϱ-cd-rules.

Remark 1.

(a) Given a set of formulae Φ such that $\psi[\varrho] \in \Phi$ and ϱ is an eventuality, the context of Φ with respect to $\psi[\varrho]$ is defined as follows

$$\mathsf{context}(\Phi, \psi[\varrho]) = \begin{cases} \Phi \setminus \{\psi[\varrho]\} & \text{if } \psi[\varrho] = \varrho \text{ or } \psi[\varrho] = \mathsf{A}(\varrho \vee \chi) \\ (\Phi \setminus \{\psi[\varrho]\}) \cup \{\chi\} & \text{if } \psi[\varrho] = \mathsf{E}(\varrho \wedge \chi) \end{cases}$$

[7] State or path formulae can appear mixed in the same set.

(b) In order to ensure the equisatisfiability of $\Phi \cup \psi[\varrho]$ and $\Phi \cup \psi[\varrho/R(\Phi, \psi[\varrho])]$ an additional side-condition is required when $\psi[\varrho]$ has the form $\mathsf{A}(\varrho \vee \gamma)$. Concretely, we apply a ϱ-cd-rule to a formula $\mathsf{A}(\varrho \vee \gamma)$ only if γ is an state formula. ∎

The rule $(\mathcal{U})_2$ is based on the fact that if $\langle \mathcal{M}, \pi \rangle \models \Delta \cup \{\circ(\varphi \,\mathcal{U}\, \psi), \neg\circ\psi\}$ is verified for some $\mathsf{CTL}^\star$-structure $\mathcal{M}$ and some infinite path $\pi \in \mathsf{paths}(\mathcal{M})$, then $\langle \mathcal{M}, \mathsf{suffix}(\pi, s_j) \rangle \models \psi$ must be verified for some $j > 1$ and $\langle \mathcal{M}, \mathsf{suffix}(\pi, s_h) \rangle \models \varphi$ must be verified for every $h \in \{1, \ldots, j-1\}$. Moreover, if Δ is true in a path $\mathsf{suffix}(\pi, s_k)$ such that $k \in \{1, \ldots, j-1\}$, and Δ is not true for every path $\mathsf{suffix}(\pi, s_\ell)$ such that $\ell \in \{k+1, \ldots, j-1\}$ then $\langle \mathcal{M}, \mathsf{suffix}(\pi, s_k) \rangle \models \Delta \cup \{\circ((\varphi \wedge \neg\Delta)\,\mathcal{U}\,\psi)\}$ and $\langle \mathcal{M}, \mathsf{suffix}(\pi, s_k) \rangle \models \Delta \cup \{\circ(\varphi \,\mathcal{U}\, \psi)\}$ are verified. Then there exists a path $\mathsf{suffix}(\pi, s_k)$ where $\Delta \cup \{\varphi \,\mathcal{U}\, \psi\}$ is true and that path is minimal in the sense that no context can be true in more that one state until the state where ψ is true. This property does not allow to postpone indefinitely the truth of ψ, provided that the number of possible contexts is finite.

As well as the above BTM-rules, the method BTM also uses the operator unnext^* to convert the labelling set $L(n)$ of a node n into another collection of labelling sets $\mathsf{unnext}^*(L(n))$ that will label as many new nodes as sets in $\mathsf{unnext}^*(L(n))$ and that intuitively represents the jump from one instant to the possible next ones. Each formula of the form $\mathsf{E}\circ\varphi$ generates a new path and an additional path is generated considering the path formulae of the form $\circ\varphi$ (note that these path formulae are not state formulae).

Given a set of formulae Φ, we denote by $\mathsf{Aunx}(\Phi)$ the set $\{\psi, \mathsf{A}\psi \mid \mathsf{A}\circ\psi \in \Phi\}$, by $\mathsf{Eunx}(\Phi)$ the set $\{\{\psi\} \mid \mathsf{E}\circ\psi \in \Phi\}$ and $\mathsf{unx}(\Phi) = \{\psi \mid \circ\psi \in \Phi\}$. Then

$$\mathsf{unnext}^*(\Phi) = \begin{cases} \{\mathsf{Aunx}(\Phi) \cup \mathsf{unx}(\Phi)\} & \text{if } \mathsf{Eunx}(\Phi) \text{ is empty} \\ \{\mathsf{Aunx}(\Phi) \cup \Omega \mid \Omega \in \mathsf{Eunx}(\Phi)\} & \text{otherwise, if } \mathsf{unx}(\Phi) \text{ is empty} \\ \{\mathsf{Aunx}(\Phi) \cup \Omega \mid \Omega \in \mathsf{Eunx}(\Phi) \cup \{\mathsf{unx}(\Phi)\}\} & \text{otherwise} \end{cases}$$

Note that, $\mathsf{unnext}^*(\Phi)$ is never empty, it contains at least one set that, in particular, could be empty.

Definition 5. *A* tableau *for a set of formulae Φ is a tuple $\mathcal{T} = (\mathsf{Nodes}, n_\Phi, L, R)$ where*

1. Nodes *is a finite non-empty set of nodes*
2. n_Φ *is a node in* Nodes*, called initial node, and it is the root of $\mathcal{T}$*
3. $L : \mathsf{Nodes} \to 2^\Gamma$ *is the labelling function where Γ is a set of formulae that contains Φ such that the the initial node is labelled by Φ, that is $L(n_\Phi) = \Phi$*
4. R *is a (successor) relation over* Nodes

and such that $\mathcal{T}$ is coherent *with respect the rules of the the system* BTM*, i.e. for each node n in* Nodes*, the labels $L(n_1), \ldots, L(n_k)$ of the collection of all nodes n_i such that $(n, n_i) \in R$ are obtained from the application of some* BTM*-rule to the label $L(n)$.* ∎

The relation R induces on Nodes a set B of *branches* of $\mathcal{T}$ formed by all the sequences $n_0, n_1, \ldots, n_k$ of nodes from Nodes such that n_0 is the initial node of $\mathcal{T}$ and $(n_i, n_{i+1}) \in R$ for all $0 \leq i < k$. A branch $b \in B$ is *maximal* iff it is not a subsequence of any other branch.

Tableaux are constructed as the refutation of the initial set of state formulae, this aim is achieved whenever the tableau is closed.

Definition 6. *A node n is* consistent *iff* $\mathbf{F} \notin L(n)$ *and there is no formula φ such that $\{\varphi, \neg\varphi\} \subseteq L(n)$. Otherwise, n is* inconsistent.[8] ∎

We classify every node that is not a leaf of a maximal branch as α-, β-, ϱ- or unnext^*-node, depending on the kind of rule that has been applied to that node. The nodes obtained by the application of a rule to a node n are called the children of n. The children of a β-node are called *disjunctive nodes* and the children of an unnext^*-node are called *key nodes*. Disjunctive and key nodes are the only nodes that could have more than one sibling (i.e. different children of the same node).

Since our tableaux combine traditional disjunctive splitting of branches with the unnext^* splitting into key nodes –which is conjunctive from the semantical point of view– we use subsets of branches, called bundles, to formulate a suitable notion of closed tableau.

Definition 7. (Bundle) *A subset Y of maximal branches in B is a* bundle *iff Y contains exactly one node from each pair of disjunctive nodes and, at the same time, all the key nodes that are children of an* unnext^**-node.* ∎

Definition 8. (Failing node) *An inconsistent node is* failing. *An α- or ϱ-node n is failing if the only node n' obtained by applying the corresponding α- or ϱ-rule to n is failing. A β-node n is failing if the two disjunctive nodes n' and n'' obtained by applying the corresponding β-rule to n are failing. An* unnext*-node n is failing if at least one of the key nodes obtained by applying the* unnext^* *operator to n is failing.* ∎

Definition 9. (Open and closed branches and bundles) *Given a tableau $\mathcal{T}_\Phi$, a branch is* closed *if it contains at least one failing node. Branches that are not closed are said to be* open. *A bundle Y is closed if every branch in Y is closed and it is open if every branch in Y is open.* ∎

4 Systematic Tableau

The correctness of the rules in BTM (which we will prove in the next section) ensures that whenever Φ is unsatisfiable, any tableau for Φ is closed. In Section 4.1 we introduce an algorithm, called $\mathcal{ST}$, that systematically constructs a tableau $\mathcal{T}_\Phi$ for any finite set of formulae Φ. This systematic tableau $\mathcal{T}_\Phi$ is closed if Φ is unsatisfiable. That is the systematic construction guarantee the refutational completeness of BTM. In addition, in Section 4.2 we show that the systematic tableau is always finite. Consequently, the algorithm $\mathcal{ST}$ is a decision procedure for $\mathsf{CTL}^\star$-satisfiability.

4.1 The Algorithm $\mathcal{ST}$

The systematic tableau construction carried out by the algorithm $\mathcal{ST}$ is based on two crucial points related to completeness (a-b) and termination (c-d).

[8] Note that, in Definition 6, the formula φ is not required to be an atom. Atomic inconsistency is strong enough in classical tableaux, although non-atomic inconsistency is more practical. In PLTL also non-atomic inconsistency is required for completeness, see [7].

a unnext^* must be applied only when there are no other applicable BTM-rules. Think about a tableau for $\{\mathsf{A}\Box p, \neg p, \circ q\}$ that starts applying unnext^*, which yields $\{q\}$.

b Context-dependent rules are essential. It is well-konwn that its context-independent versions are not strong enough for completeness.

c A systematic use of the context-dependent rules that prevents that infinitely many formulae should be generated by means of context formulae.

d Detection of when an open branch is sufficiently extended

Consequently, (1) we deal with the notions of stage, cyclic branch, saturated set and fulfilling branch, for detecting when an open branch is sufficiently extended and (2) the algorithm $\mathcal{ST}$ applies the corresponding context-dependent rule to exactly one eventuality (if there is some) between each two consecutive applications of unnext^*. For that, the algorithm $\mathcal{ST}$ uses a function d that associates with each node so-called distinguished eventuality. The ϱ-cd-rules (see Fig. 2) can only be applied to the distinguished eventuality. Along the construction of the systematic tableau, the function d associates with every node n one of the following three possible sets of formulae:

1. the empty set
2. a non-elementary singleton $\{\chi\}$ where χ is of the form $\varphi \,\mathcal{U}\, \psi$, $\Diamond\varphi$ or $\neg\Box\varphi$
3. an elementary singleton of the form $\{\circ(\varphi \,\mathcal{U}\, \psi)\}$.

Case 1 means that no eventuality is distinguished. In 2, d yields the set containing the distinguished formula to which $(\,\mathcal{U}\,)_2$, $(\Diamond)_2$ or $(\neg\Box)_2$ will be applied. Case 3 means that after the application of $(\,\mathcal{U}\,)_2$, $(\Diamond)_2$ or $(\neg\Box)_2$ the new distinguished formula is $\varphi \,\mathcal{U}\, \psi$. At the beginning, d associates the empty set with the initial node. When the set of all possible different labels is finite, any infinite branch must contain infinitely many different nodes with the same label. In particular, when a repetition arises in an open branch of the form $n_0, n_1, \ldots, n_{j-1}, n_j, \ldots, n_k$ (i.e., when $L(n_k) = L(n_{j-1})$ for some $0 < j \leq k$), then an infinite branch of the form $n_0, n_1, \ldots, n_{j-1}, n_j, \ldots, n_k, n_j, \ldots, n_k, \ldots$ can be obtained. In fact, this will be a cyclic branch that will be finitely represented. If $b = n_0, n_1, \ldots, n_k$ is an open branch such that $L(n_k) = L(n_{j-1})$ for some $0 < j \leq k$, then b is cyclic and we define $\mathsf{path}(b) = n_0, n_1, \ldots, n_{j-1} \cdot \langle n_j, n_{j+1}, \ldots, n_k\rangle^\omega$ and $\mathsf{cycle}(b) = n_j, n_{j+1}, \ldots, n_k$. In other words, we consider that the implicit successor relation on b is extended with $n_k R n_j$. If a branch b' is not cyclic then $\mathsf{path}(b') = b'$. The last node n_k whose label appears previously in the branch is intentionally added to the branch because this repetition is what we use for detecting the loop. Every branch (cyclic or not) of a coherent pre-tableau can be seen as divided into *stages* according to the applications of the operator unnext^*. In other words, a stage is a sequence of consecutive nodes between two consecutive applications of the unnext^* operator.

Definition 10. *Given a branch b, every maximal subsequence $n_i, n_{i+1}, \ldots, n_j$ of $\mathsf{path}(b)$ such that n_ℓ is not a key node for every $i < \ell \leq j$, is called a* stage*. We denote by $\mathsf{stages}(b)$ the* sequence of all stages *of a branch b. The successor relation on $\mathsf{stages}(b)$ is induced by the successor relation on $\mathsf{path}(b)$. That is, if s and s' are respectively stages $n_0, \ldots n_i$ and $n'_0, \ldots n'_j$ in $\mathsf{path}(b)$ then sRs' whenever $n_i R n'_0$. Hence, if b is a cyclic sequence of nodes, then $\mathsf{stages}(b)$ is a cyclic sequence of stages.* ∎

Example 11. Consider a cyclic branch $b = n_1, n_2, n_3, n_4, n_5$ such that $L(n_5) = L(n_3)$. Then, $\mathsf{path}(b) = n_1, n_2, n_3 \cdot \langle n_4, n_5 \rangle^\omega$. Let us suppose that $L(n_2) = \mathsf{unnext}(L(n_1))$ and $L(n_5) = \mathsf{unnext}(L(n_4))$. Then, $\mathsf{stages}(b)$ is formed by three stages: $s_1 = \langle n_1 \rangle$, $s_2 = \langle n_2, n_3, n_4 \rangle$ and $s_3 = \langle n_5, n_4 \rangle$. Therefore, the induced relation R on $\mathsf{stages}(b)$ is given by $s_1 R s_2$, $s_2 R s_3$ and $s_3 R s_3$. Hence, $\mathsf{path}(\mathsf{stages}(b)) = s_1, s_2 \cdot \langle s_3 \rangle^\omega$. ∎

With a slight abuse of notation, the labelling function L is extended from nodes to stages in the natural way. That is, for any stage s: $L(s) = \bigcup_{n \in s} L(n)$.

Definition 12. *Let S be a sequence of stages, $s \in S$ and $\varphi \mathcal{U} \psi \in L(s)$, we say that $\varphi \mathcal{U} \psi$ is fulfilled in S iff there exists s' such that $s R^* s'$ and $\psi \in L(s')$. A sequence S of stages is* fulfilling *iff for all $s \in S$ every $\varphi \mathcal{U} \psi \in L(s)$ is fulfilled in S. A branch b is fulfilling iff the sequence $\mathsf{path}(\mathsf{stages}(b))$ is fulfilling.* ∎

Input: A finite set of formulae Φ
Output: An expanded tableau $\mathcal{T}_\Phi = (\mathsf{Nodes}, n_\Phi, L, R)$ for Φ

```
1  Nodes := {n_Φ}; L := {(n_Φ, Φ)}; R := { }; d := {(n_Φ, { })}
2  while not empty?(unmarked_branches(B)) loop
3     choose b ∈ unmarked_branches(B)
4     n := last_node(b)
5     if empty?(d(n)) then fairly_select_eventuality(T_Φ, d) end if
6     if elementary?(L(n))
7        then apply_unnext*(T_Φ, d)
8        else choose a non-elementary γ such that γ ∈ L(n) or φ[γ] ∈ L(n)
9             if d(n) = {γ} then apply_ϱ_cd_rule(γ, T_Φ, d)
10                          else apply_αβϱ_rule(γ, T_Φ, d)
11            end if
12    end if
13 end loop
```

Fig. 3. The Algorithm $\mathcal{ST}$ for Systematic Tableau Construction

The concept of a fulfilling branch together with the following concept of $\alpha\beta\varrho$-saturated stage is crucial for determining when open branches are sufficiently expanded in order to be part of a bundle that is able to describe a model.

Definition 13. *A stage s is* $\alpha\beta\varrho$-saturated *if, and only if, for every $\varphi \in L(s)$:*

1. *If φ is an α-formula then $A(\varphi) \subseteq L(s)$*
2. *If φ is a β-formula then $B_1(\varphi) \subseteq L(s)$ or $B_2(\varphi) \subseteq L(s)$*
3. *Otherwise, if $\psi_1, \psi_2, \ldots, \psi_\ell$ (with $\ell \geq 1$) are subformulae of φ that are ϱ-formulae, then $\varphi[\psi_j / R(\psi_j)] \in L(s)$ or $\varphi[\psi_j / R(\psi_j, \mathsf{context}(\Phi, \varphi[\psi_j]))] \in L(s)$, for some $j \in \{1, \ldots, \ell\}$ and some $n \in s$ such that $\varphi[\psi_j] \in L(n)$.* ∎

Now, we give a sufficient condition to consider that a branch is (sufficiently) expanded. This condition can be syntactically checked. Later, after the algorithm $\mathcal{ST}$ has been presented, we will refine this syntactic condition to a simpler one.

Definition 14. *An open branch b is* expanded *if and only if b is fulfilling, cyclic and each stage $s \in \mathsf{stages}(b)$ is $\alpha\beta\varrho$-saturated. An open bundle Y is expanded if every branch in Y is expanded.* ∎

For example, an expanded branch of a coherent pre-tableau for the set of well-formed $\mathsf{CTL}^\star$ formulae $\Psi = \{\mathsf{A}(r\,\mathcal{U}\,p)\}$ can be formed by the sequence of stages $b = s_0, s_1, s_2$ where $L(s_0) = \{\mathsf{A}(r\,\mathcal{U}\,p), \mathsf{A}(p \vee (r \wedge \circ(r\,\mathcal{U}\,p))), p \vee \mathsf{A}(r \wedge \circ(r\,\mathcal{U}\,p)), p\}$ and $L(s_1) = L(s_2) = \{\ \}$. Actually the branch b is fulfilling, cyclic and $\alpha\beta\varrho$-saturated, hence it is expanded. The set $Y = \{b\}$ is an open expanded bundle that describes a model for Ψ.

It is worth to note that expanded branches can be enlarged. Enlargement of expanded branches will be used later in order to ensure the construction of fulfilling branches without checking directly whether a branch is fulfilling.

Definition 15. *An* expanded tableau *is a tableau where every maximal branch is either expanded or closed. An expanded tableau is open iff there exists an open bundle. Otherwise it is closed.* ∎

Note that to check whether an expanded tableau $\mathcal{T}_\Phi$ contains an open bundle it suffices to check whether $\mathcal{T}_\Phi$ contains an open maximal branch.

The algorithm $\mathcal{ST}$, which is depicted by a while-program in Figure 3, provides a proof search procedure for automated deduction. First, it initializes the tableau $\mathcal{T}_\Phi$ with the initial node labeled by the input set of formulae Φ and the function d for this initial node as empty (not eventuality is selected). Then it iterates a process that nondeterministically selects, at each step, a maximal branch b to be extended (enlarged or split) depending on the last node n in b. At the beginning of each iteration step, it checks if there is a distinguished eventuality. If no eventuality is already distinguished in n and $L(n)$ contains some eventuality to which a ϱ-cd-rule can be applied[9], then the procedure $fairly\_select\_eventuality(\mathcal{T}_\Phi, d)$ selects one and mark it as the distinguished eventuality of node n. That is, whenever $d(n)$ is empty and $L(n)$ contains at least one selectable eventuality, it updates $d(n)$ with a singleton $\{\varphi\}$ such that $\varphi \in L(n)$ is an eventuality . Otherwise, $d(n)$ remains the empty set. If the node contains more than one eventuality, the selection should be *fair*, in the sense that no eventuality could remain non-distinguished indefinitely. Then, three different extensions can be made. If every formula in $L(n)$ is elementary, hence the procedure $apply\_unnext^*$ applies unnext^* to the node n and, also, assigns the set $\{\chi \mid \circ\chi \in d(n)\}$ to $d(n')$ for every new node n'. Otherwise, it chooses one non-elementary formula γ such that $\gamma \in L(n)$ or $\varphi[\gamma] \in L(n)$ and proceeds depending on the function d as follows:

- If γ is the distinguished formula then there is a non-empty set of formulae, namely Ω, of the form $\varphi[\gamma]$ in $L(n)$. Then, it simultaneously applies the ϱ-cd-rule (R) that corresponds to the distinguished eventuality γ to all the occurrences of γ in Ω to which (R) is applicable. In this case, the procedure $apply\_\varrho\_cd\_rule$ associates, as d value, the new next-formula that includes the negated context as subformulae.
- Otherwise, if γ is not the distinguished formula, then it applies the α-, β- or ϱ-rule that corresponds according to the form of γ. This is the task of the procedure

[9] Recall the restriction of Remark 1(b)

$apply\_\alpha\beta\varrho\_rule$, which also assigns $d(n)$ to $d(n')$ for every new node n' generated by the rule application.

Finally, the algorithm $\mathcal{ST}$ terminates when every maximal branch is marked either as closed or as expanded. When it is detected that a maximal branch is either closed or expanded that branch is consequently marked. The procedure $unmarked\_branches$ yields the maximal branches that are not marked yet.

4.2 Termination of the Algorithm $\mathcal{ST}$

In this section we show that the systematic tableau construction terminates for any set of formulae Φ. The tableau method BTM satisfies what we call the *weak analytic superformula property* (WASP), i.e. for every finite set of formulae Φ, there exists a finite set $\mathsf{clo}(\Phi)$ (closure of Φ) that contains all the formulae that may occur in any systematic tableau for Φ. Termination follows from finiteness of this closure and fairness in distinguishing eventualities.

For the sake of simplicity, along this subsection, we consider until formulae as the only eventualities in the proofs, since the other eventualities are especial cases of until formulae.

As an initial step towards the definition of the closure of a set of formulae, we first define the notion of pre-closure.

Definition 16. *Let* $\mathsf{subf}(\Phi)$ *denote the set of all the subformulae of the formulae in* Φ *and their negations. Let* $\mathsf{superf}(\Phi)$ *denote the smallest set that extends* $\mathsf{subf}(\Phi)$ *verifying that for any* $\psi \in \mathsf{superf}(\Phi)$ *every subformula of* ψ *is in* $\mathsf{superf}(\Phi)$ *and for every* $\psi[\varrho] \in \mathsf{superf}(\Phi)$ *the formula* $\psi[\varrho/R(\varrho)] \in \mathsf{superf}(\Phi)$*. Then, the preclosure of* Φ*,* $\mathsf{preclo}(\Phi)$*, is the set* $\mathsf{superf}(\Phi) \cup \{\mathsf{A}\psi, \mathsf{E}\psi \mid \psi \in \mathsf{superf}(\Phi)\}$. ■

Note that $\mathsf{preclo}(\Phi)$ cannot be used as closure only because it does not capture formulae generated by the application of the ϱ-cd-rules $(\mathcal{U})_2$,$(\Diamond)_2$ and $(\neg\Box)_2$. In order to capture these superformulae, we define the set of conjunctions of negated contexts. Given a set $\Psi = \{\psi_1, \ldots, \psi_n\}$ we denote by $\bigwedge \Psi$ the formula $\psi_1 \wedge \ldots \wedge \psi_n$ and, in particular, when Ψ is empty, $\bigwedge \Psi$ is the constant $\mathbf{T}$.

Definition 17. *Let* $\mathsf{negctx}(\Phi) = \{\neg\Delta \mid \Delta \subseteq \mathsf{preclo}(\Phi)\}$ *be the set of all possible* negated contexts*. The set* $\mathsf{conj}(\Phi)$ *is formed by all the possible conjunctions of formulae in* $\mathsf{negctx}(\Phi)$*.*

$$\mathsf{conj}(\Phi) = \{\bigwedge \Gamma \mid \Gamma \subseteq \mathsf{negctx}(\Phi)\}$$

In particular, $\mathbf{F} \in \mathsf{negctx}(\Phi)$ *and* $\mathbf{F}, \mathbf{T} \in \mathsf{conj}(\Phi)$*, since* $\mathbf{F}$ *and* $\mathbf{T}$ *are respectively the disjunction and the conjunction of the empty set of formulae.* ■

Note that, by definition, in the conjunctions of $\mathsf{conj}(\Phi)$ every element of $\mathsf{negctx}(\Phi)$ occurs at most once.

Now we give some results about the systematic tableau $\mathcal{T}_\Phi$ that the algorithm $\mathcal{ST}$ constructs for any set of formulae Φ. This results are useful to define a finite closure and consequently to ensure the termination of the construction for every Φ.

Proposition 18. *If $\{\varphi, \neg\varphi\} \subseteq L(s)$ for some stage s in a branch b of $\mathcal{T}_\Phi$, then every maximal branch of $\mathcal{T}_\Phi$ prefixed by b is closed.*

Proof. By structural induction on φ. It is easy to see that the application of BTM-rules to two complementary formulae that belong to the same stage, but not necessarily to the same node, should generate complementary constituents until they occur in the same node or, at most, they become elementary. ∎

The following proposition states that non-satisfied undistinguished eventualities are kept in branches at least until they are fulfilled or they become distinguished.

Proposition 19. *Let b be a branch[10] of $\mathcal{T}_\Phi$, and $s_0, s_1, s_2, \ldots, s_k$ be any initial subsequence of $\mathsf{path}(\mathsf{stages}(b))$. If $\varphi\,\mathcal{U}\,\psi \in L(s_i)$ for some $0 \leq i \leq k$, $\varphi\,\mathcal{U}\,\psi$ is not distinguished in $s_i, \ldots, s_k$ and $\psi \notin L(s_i) \cup \ldots \cup L(s_k)$, then $\{\varphi, \neg\psi, \circ(\varphi\,\mathcal{U}\,\psi)\} \subseteq L(s_j)$ for all $i \leq j \leq k$.*

Proof. By the construction of $\mathcal{T}_\Phi$, since non-distinguished eventualities are handled by procedure $apply\_\alpha\beta\varrho\_cd\_rule$ using the rule $(\,\mathcal{U}\,)_1$. ∎

Next, we give a more detailed description of the syntactic form of the formulae appearing in sequences of stages where a distinguished eventuality remains unfulfilled. At each stage, there is exactly one distinguished eventuality and exactly one node to which the procedure $apply\_\beta cd\_rule$ is applied. We also call this node the *distinguished node* of that stage.

Proposition 20. *Let b be a branch[10] of $\mathcal{T}_\Phi$, let $s_0, s_1, s_2, \ldots, s_k$ be any initial subsequence of $\mathsf{path}(\mathsf{stages}(b))$ and $\varphi\,\mathcal{U}\,\psi \in \mathsf{subf}(\Phi)$ such that i is the least natural number such that $d(n) = \{\varphi\,\mathcal{U}\,\psi\}$ for some $n \in s_i$. Then, if $\psi \notin L(s_i) \cup \ldots \cup L(s_k)$ then for all $1 \leq \ell \leq k - i$:*

$$\{\delta_\ell, \circ(\delta_{\ell+1}\,\mathcal{U}\,\gamma_{\ell+1})\} \subseteq L(s_{i+\ell})$$

where $\delta_0 = \varphi$, $\gamma_0 = \psi$ and $\delta_{\ell+1} = \delta_\ell \wedge \chi$ and $\gamma_{\ell+1} = \gamma_\ell \wedge \chi$ for some $\chi \in \mathsf{negctx}(\Phi)$. Moreover, if $\delta_\ell = \bigwedge \Gamma$ for some Γ such that $\chi \in \Gamma$ then every maximal branch of $\mathcal{T}_\Phi$ prefixed by $s_0, \ldots, s_{i+\ell}$ is closed and every maximal branch that contains the sibling m' of the first node $m \in s_{i+\ell}$ such that $\{\delta_\ell, \circ(\delta_{\ell+1}\,\mathcal{U}\,\gamma_{\ell+1})\} \subseteq L(m)$, is closed.

Proof. On the one hand, by construction of $\mathcal{T}_\Phi$ and induction on ℓ, the procedure $apply\_\varrho\_cd\_rule$ yields two branches such that each branch either contains the formulae in the set $\{\delta_\ell, \circ(\delta_{\ell+1}\,\mathcal{U}\,\gamma_{\ell+1})\}$ or contains the formula $\gamma_{\ell+1}$. Note that if $d(n) = \{\circ(\delta_{\ell+1}\,\mathcal{U}\,\gamma_{\ell+1})\}$ for some $n \in s_{i+\ell}$, then $d(n') = \{\delta_{\ell+1}\,\mathcal{U}\,\gamma_{\ell+1}\}$ for the first node $n' \in s_{i+\ell+1}$. Therefore, $\delta_0 = \varphi$, $\gamma_0 = \psi$ and for all $j > 0$: $\delta_j = \delta_{j-1} \wedge \neg\Delta_{j-1}$ and $\gamma_j = \gamma_{j-1} \wedge \neg\Delta_{j-1}$ where $\neg\Delta_{j-1} \in \mathsf{negctx}(\Phi)$ and Δ_{j-1} is the context $L(n) \setminus d(n)$ of the distinguished node n of the stage s_{i+j-1}. Hence, by induction on ℓ, $\delta_\ell \in \mathsf{conj}(\Phi)$ holds for all $0 \leq \ell \leq k - i$.

On the other hand, since χ is the negation of the context of the distinguished node $n \in s_{i+\ell}$, if $\delta_{\ell+1} = \delta_\ell \wedge \chi$ and $\delta_\ell = \bigwedge \Gamma$ for some Γ such that $\chi \in \Gamma$, then every

[10] The branch b could be cyclic or not, so that $\mathsf{path}(\mathsf{stages}(b))$ could respectively be infinite or finite.

branch prefixed by $s_0, \ldots, s_{i+\ell}$ contains at the same stage (possibly at different nodes) $\{\zeta, \neg\zeta\}$ for some formula ζ. Hence, by Proposition 18, every maximal branch prefixed by $s_0, \ldots, s_{i+\ell}$ is closed. By the same reasoning every maximal branch that contains the sibling m' of the first node $m \in s_{i+\ell}$ such that $\{\delta_\ell, \circ(\delta_{\ell+1} \,\mathcal{U}\, \gamma_{\ell+1})\} \subseteq L(m)$, is closed. ∎

Our notion of closure is based on the following results.

Corollary 21. *Every distinguished eventuality in a cyclic branch of $\mathcal{T}_\Phi$ is fulfilled.*

Proof. Every application of the rule $(\,\mathcal{U}\,)_2$ generates a disjunction of the form $\gamma_\ell \vee (\delta_\ell \wedge \circ(\delta_{\ell+1} \,\mathcal{U}\, \gamma_{\ell+1}))$. By Proposition 20, sequences of nodes that include different disjuncts of the second type generated from the same initial eventuality cannot form a cycle since the presence of the formulae δ_ℓ makes impossible the existence of a loop in such a sequences. ∎

Corollary 22. *In formulae of the form $\delta_\ell \,\mathcal{U}\, \gamma_\ell$ generated by the ϱ-cd-rule $(\,\mathcal{U}\,)_2$ a context can only have two appearances.*

Proof. Due the possibility of having more than one instance of a distinguished eventuality –since it can appear as non-state formula and besides as subformula of an A-formula– by Proposition 20, the presence of the formulae γ_ℓ makes impossible the existence of formulae of the form $\delta_\ell \,\mathcal{U}\, \gamma_\ell$ containing more than two appearances of a given negated context. Without the presence of the formulae γ_ℓ it would be possible to have formulae of the form $\delta_\ell \,\mathcal{U}\, \gamma_\ell$ inside an A-formula, containing in δ_ℓ more than two appearances of a given negated context. ∎

Consequently, BTM satisfies the WASP with respect to the following notion of closure:

$$
\begin{aligned}
\mathsf{clo}(\Phi) = {} & \mathsf{preclo}(\Phi) \cup \mathsf{conj}(\Phi) \\
& \cup \{(\varphi \wedge \chi) \,\mathcal{U}\, (\psi \wedge \chi), \circ((\varphi \wedge \chi) \,\mathcal{U}\, (\psi \wedge \chi)) \mid \varphi \,\mathcal{U}\, \psi \in \mathsf{subf}(\Phi) \\
& \quad \text{and } \chi \in \mathsf{conj}(\Phi) \text{ or } \chi = \gamma_1 \wedge \gamma_2 \text{ where } \gamma_1, \gamma_2 \in \mathsf{conj}(\Phi)\} \\
& \cup \{\chi \,\mathcal{U}\, (\varphi \wedge \chi), \circ(\chi \,\mathcal{U}\, (\varphi \wedge \chi)) \mid \Diamond\varphi \in \mathsf{subf}(\Phi) \\
& \quad \text{and } \chi \in \mathsf{conj}(\Phi) \text{ or } \chi = \gamma_1 \wedge \gamma_2 \text{ where } \gamma_1, \gamma_2 \in \mathsf{conj}(\Phi)\} \\
& \cup \{\chi \,\mathcal{U}\, (\neg\varphi \wedge \chi), \circ(\chi \,\mathcal{U}\, (\neg\varphi \wedge \chi)) \mid \neg\Box\varphi \in \mathsf{subf}(\Phi) \\
& \quad \text{and } \chi \in \mathsf{conj}(\Phi) \text{ or } \chi = \gamma_1 \wedge \gamma_2 \text{ where } \gamma_1, \gamma_2 \in \mathsf{conj}(\Phi)\}
\end{aligned}
$$

Note that, γ_1 and γ_2 are enough to represent the unique possible repetition of a negated context. Hence, $L(n) \subseteq \mathsf{clo}(\Phi)$ holds for all node n in $\mathcal{T}_\Phi$, by means of Corollary 21, Corollary 22, Remark 2 and Proposition 19. In addition, it is easy to see that $\mathsf{clo}(\Phi)$ is finite for any finite set of formulae Φ.

Proposition 23. *If Φ is a finite set of formulae, then $\mathsf{clo}(\Phi)$ is also finite.*

Proof. It is easy to see that, if $|\mathsf{preclo}(\Phi)| = n$ then $|\mathsf{negctx}(\Phi)| \in O(2^n)$. As a consequence $|\mathsf{conj}(\Phi)|, |\mathsf{clo}(\Phi)| \in O(2^{O(2^n)})$. ∎

This fact jointly with the fairness of $fairly\_select\_eventuality$, allows us to ensure that the algorithm in Figure 3 finitely computes an expanded tableau for any input.

Lemma 24. *The algorithm in Figure 3, for any input Φ, terminates with an expanded tableau in $\mathcal{T}_\Phi$.* ∎

The particular strategy followed by the algorithm $\mathcal{ST}$ allows us to formulate the following refinement of the sufficient conditions for being an expanded branch (see Definition 14).

Proposition 25. *Given an open branch b of $\mathcal{T}_\Phi$, if b satisfies the following two conditions:*
(i) *b is cyclic*
(ii) *for every eventuality $\gamma \in \mathsf{preclo}(\Phi)$ such that $\gamma \in L(n)$ for some $n \in \mathsf{cycle}(b)$, there exists some $n' \in \mathsf{cycle}(b)$ such that $d(n') = \{\gamma\}$*
then b is an expanded branch.

Proof. By Proposition 19, non-distinguished unfulfilled eventualities are preserved from one stage to its successor. In addition, by Corollary 21, every distinguished eventuality in a cyclic branch is fulfilled. Hence, by condition (ii), every eventuality from $\mathsf{preclo}(\Phi)$ that occurs in b should be distinguished once and, hence, should be fulfilled. ∎

Consequently, we use Proposition 25 to refine the implementation of the procedure $unmarked\_branches$.

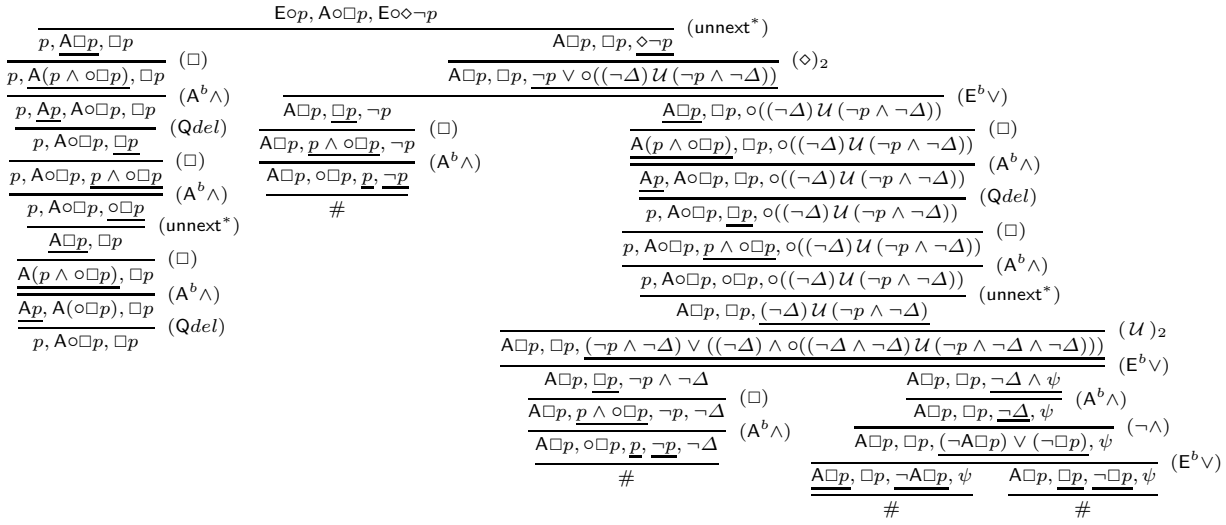
Remark 2. Whenever a branch b satisfies the conditions (i) and (ii) of Proposition 25, the procedure $unmarked\_branches$ considers b to be marked as expanded. ∎

4.3 Tableau Construction Examples

In this section we present two closed systematic tableaux. The rule applied in each step is indicated at the right-hand side and the formula or formulae to which the rules apply are underlined. In the steps where the unnext^* operator is applied, the formulae are not underlined since this operator applies to all the formulae of the considered set of formulae.

Example 1. In Figure 4 there is an example of a systematic closed tableau constructed by the algorithm $\mathcal{ST}$. Note that its left-most branch is cyclic and its leaf is not inconsistent and gives a model for $\{p, \mathsf{A}\Box p, \Box p\}$. The other four branches contain inconsistent leaves. The tableau contains four bundles and each of them contains two branches, the left-most one and one of the remaining four branches. Every bundle is closed and that ensures the unsatisfiability of the set in the root. ∎

Example 2. While testing the technique on a variety of $\mathsf{CTL}^\star$ formulae, we have found particularly useful those formulae that are characteristic for this logic, and are related to the limit closure property. In other words, their validity is due to the limit closure property. One of such formulae is the unsatisfiable formula $\mathsf{A}\Diamond((\circ\neg p) \wedge (\mathsf{E}\circ p))$. In Figure 5 there is a systematic closed tableau for this formula. In this example all the branches contain inconsistent leaves and each branch constitute a bundle. Note that the admissible rule $(Sbm\vee)_a$, that given two formulae of the form $\mathsf{A}^b(\varphi \vee \psi)$ and $\mathsf{A}^b(\varphi)$ deletes (by the subsumption principle) the formula $\mathsf{A}^b(\varphi \vee \psi)$, is used. Admissible rules are sound rules that do not add deduction power to the system (more about admissible rules can be found in [8] and [7]). ∎



where $\Delta = \{\mathsf{A}\Box p, \Box p\}$, $\neg\Delta = \neg((\mathsf{A}\Box p) \wedge (\Box p))$ and $\psi = \circ((\neg\Delta \wedge \neg\Delta)\,\mathcal{U}\,(\neg p \wedge \neg\Delta \wedge \neg\Delta))$

Fig. 4. An example of systematic tableau for $\{\mathsf{E}\circ p, \mathsf{A}\circ\Box p, \mathsf{E}\circ\Diamond\neg p\}$

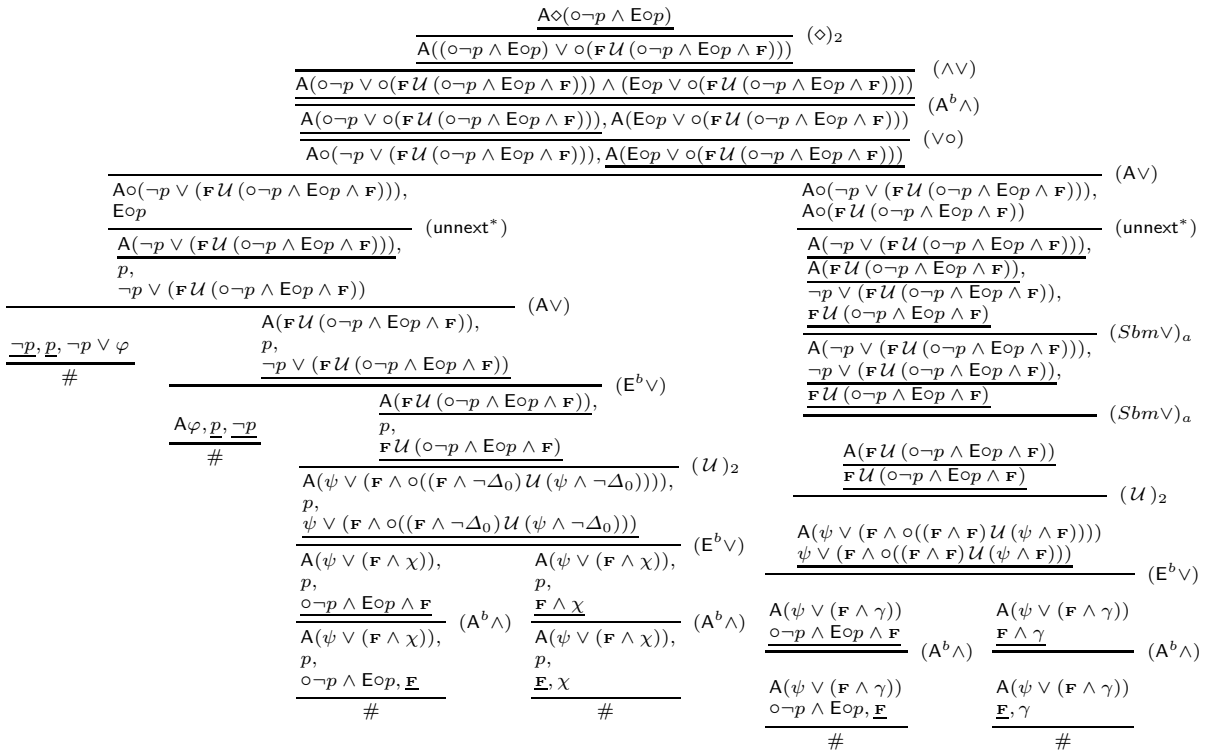


Fig. 5. An example of systematic tableau for $\{\mathsf{A}\Diamond((\circ\neg p) \wedge (\mathsf{E}\circ p))\}$

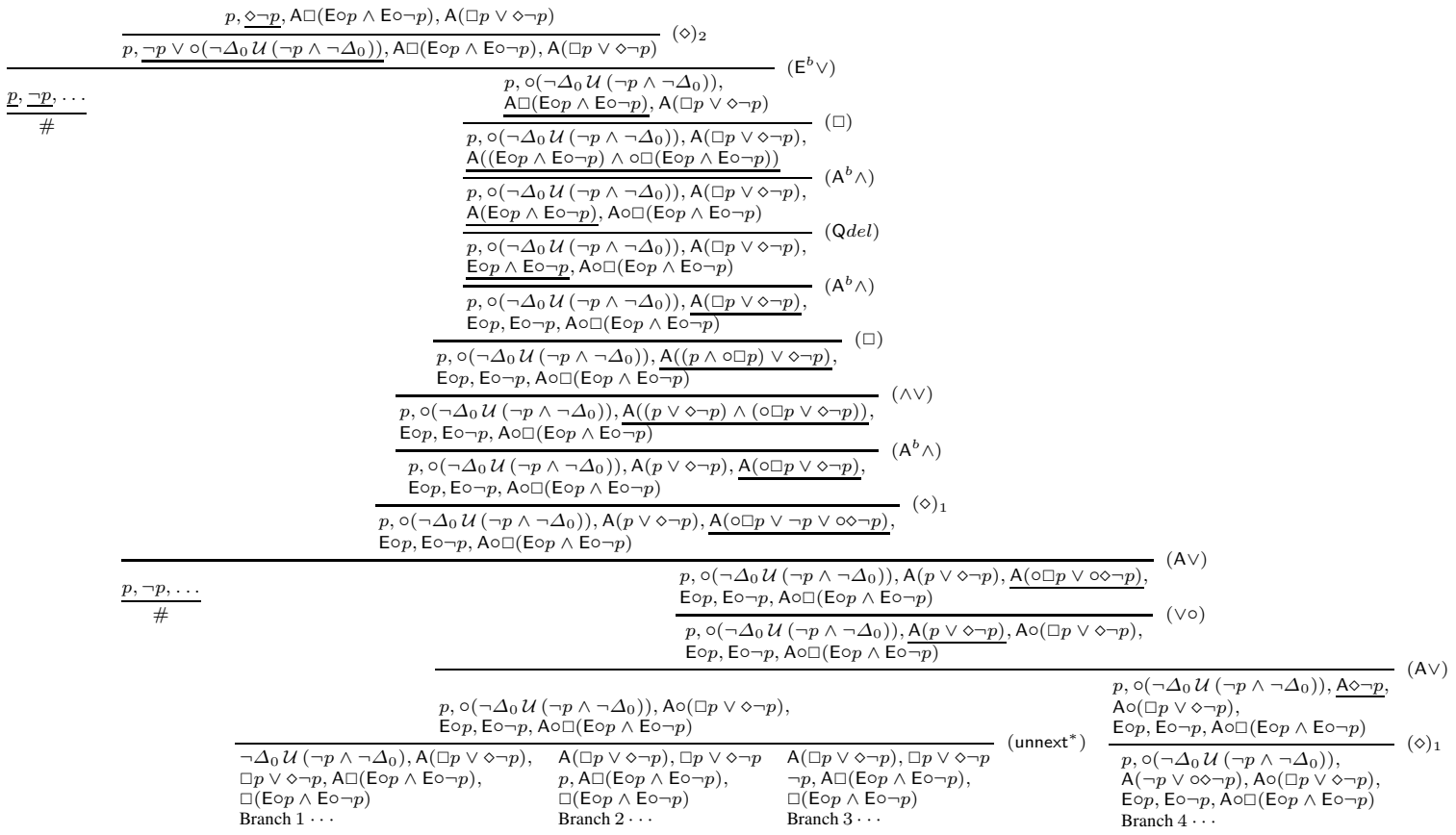


Fig. 6. An example of systematic tableau for $\{p, \Diamond\neg p, \mathsf{A}\Box(\mathsf{E}\circ p \wedge \mathsf{E}\circ\neg p), \mathsf{A}(\Box p \vee \Diamond\neg p)\}$

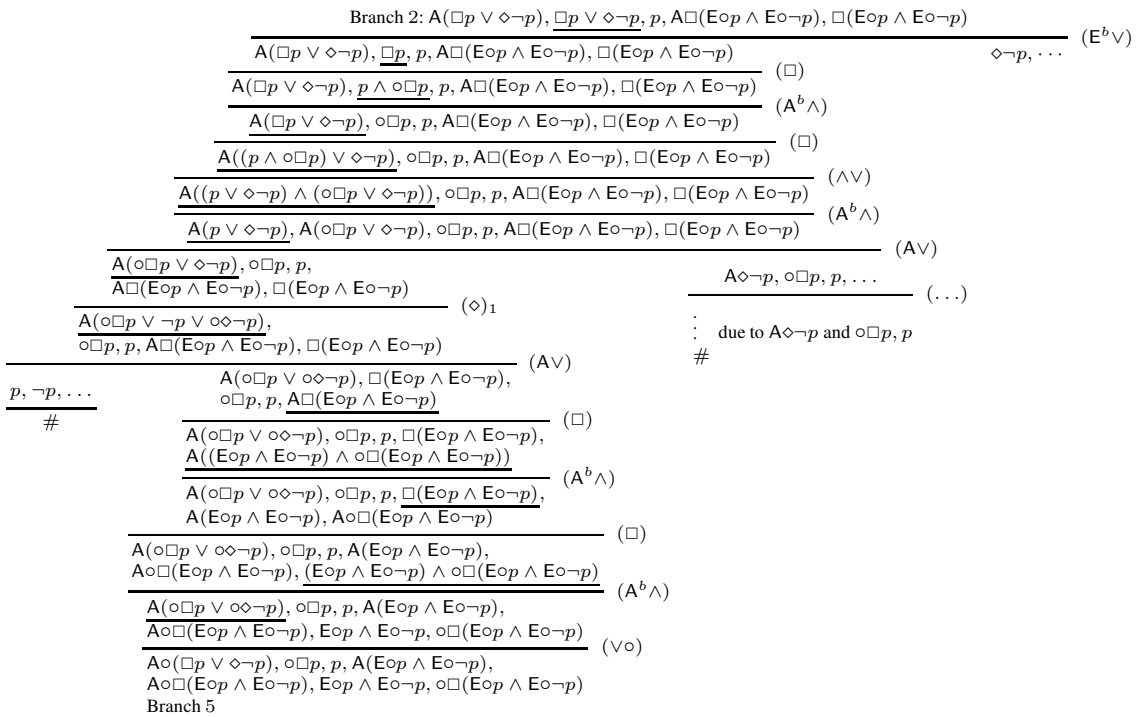


Fig. 7. Extension of branch 2 in Fig. 6

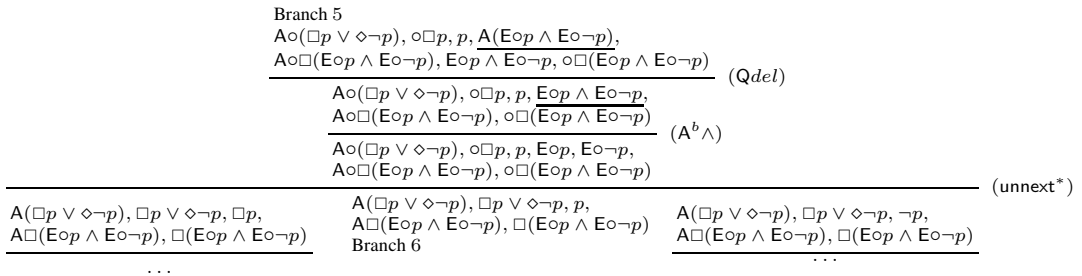


Fig. 8. Extension of branch 5 in Fig. 7

Example 3. In this example we deal with the satisfiable set of formulas

$$\Phi = \{p, \Diamond\neg p, \mathsf{A}\Box(\mathsf{E}\circ p \wedge \mathsf{E}\circ\neg p), \mathsf{A}(\Box p \vee \Diamond\neg p)\}$$

Since a systematic tableau for Φ is very big, we only show the construction of an open branch. The construction is spread in Figures 6, 7 and 8. The construction of branch 2 in Fig. 6 is followed in Fig. 7 and the construction of branch 5 in Fig. 7 is followed in Fig. 8. The last set of formulas in the branch 6 in Fig. 8 is the same set that appears at the beginning of Fig. 7, so that we have a cyclic branch. The development of branches 1 and 3 is similar to the development of branch 2. In order to obtain an open bundle based on branch 2, branches 1 and 3 are also necessary. ∎

5 Correctness

5.1 Soundness

A tableau method is *sound* if, whenever a closed tableau exists for Φ, then Φ is unsatisfiable. The soundness of BTM is a direct consequence of the fact that the source and target of every BTM-rule are equi-satisfiable and the operator unnext^* preserves satisfiability.

Lemma 26. *For every set of formulae Φ, any α-formula γ, any β-formula ψ, any ϱ-formula ϱ and any ϱ-cd-formula ϱ':*

1. *$\Phi \cup \{\gamma\}$ is satisfiable iff $\Phi \cup A(\gamma)$ is satisfiable*
2. *$\Phi \cup \{\psi\}$ is satisfiable iff $\Phi \cup B_1(\psi)$ or $\Phi \cup B_2(\psi)$ is satisfiable.*
3. *$\Phi \cup \{\varphi[\varrho]\}$ is satisfiable iff $\Phi \cup \{\varphi[\varrho/R(\varrho)]\}$ is satisfiable.*
4. *$\Phi \cup \{\varphi[\varrho']\}$ is satisfiable iff $\Phi \cup \{\varphi[\varrho'/R(\varrho', \mathsf{context}(\Phi, \varphi[\varrho']))]\}$ is satisfiable.*
5. *If Φ is satisfiable then every $\Psi \in \mathsf{unnext}^*(\Phi)$ is satisfiable.*

Proof. The right-to-left implications are trivial. The left-to-right implications of items 1, 2, 3, and 5, are routine to check. For the item 4, we will study the three possible cases: $\varphi\,\mathcal{U}\,\psi$, $\mathsf{E}((\varphi\,\mathcal{U}\,\psi) \wedge \chi)$ and $\mathsf{A}((\varphi\,\mathcal{U}\,\psi) \vee \chi)$.

First, let us assume that there exists a CTL^*-structure $\mathcal{M}$ and $\pi \in \mathsf{fullpaths}(\mathcal{M})$ such that $\langle \mathcal{M}, \pi\rangle \models \Phi \cup \{\varphi\,\mathcal{U}\,\psi\}$. The we built a CTL^*-structure $\mathcal{M}'$ and $\pi' \in \mathsf{fullpaths}(\mathcal{M}')$ such that $\langle \mathcal{M}', \pi'\rangle \models \Phi \cup \{\psi \vee (\varphi \wedge \circ((\varphi \wedge \neg\Phi)\,\mathcal{U}\,(\psi \wedge \neg\Phi)))\}$ as follows. If $\langle \mathcal{M}, \pi\rangle \models \psi$, then, $\mathcal{M}' = \mathcal{M}$ and π' is the path π. Otherwise, if $\langle \mathcal{M}, \pi\rangle \not\models \psi$, then $\langle \mathcal{M}, \pi\rangle \models \circ(\varphi\,\mathcal{U}\,\psi)$. Let x be the least $z \geq 1$ such that $\langle \mathcal{M}, \mathsf{suffix}(\pi, s_z)\rangle \models \psi$. Then there are two cases depending on $\langle \mathcal{M}, \mathsf{suffix}(\pi, s_z)\rangle \models \Phi$ or not. In the latter case, $\langle \mathcal{M}, \mathsf{suffix}(\pi, s_z)\rangle \models \psi \wedge \neg\Phi$. Thus, by considering y to be the greatest h such that $0 \leq h < x$ and $\langle \mathcal{M}, \mathsf{suffix}(\pi, s_h)\rangle \models \Phi \cup \{\varphi\,\mathcal{U}\,\psi\}$ (note that $h = 0$ satisfies this), we have that

$$\langle \mathcal{M}, \mathsf{suffix}(\pi, s_y)\rangle \models \Phi \cup \{\varphi, \neg\psi, \circ((\varphi \wedge \neg\Phi)\,\mathcal{U}\,(\psi \wedge \neg\Phi))\}.$$

Therefore $\mathcal{M}'$ is the sub-structure of $\mathcal{M}$ whose root is s_y and $\pi' = \mathsf{suffix}(\pi, s_y) \in \mathsf{fullpaths}(\mathcal{M}')$. In the former case, $\langle \mathcal{M}, \mathsf{suffix}(\pi, s_z)\rangle \models \Phi \wedge \psi$, so that $\mathcal{M}'$ is the sub-structure of $\mathcal{M}$ whose root is s_z and $\pi' = \mathsf{suffix}(\pi, s_z) \in \mathsf{fullpaths}(\mathcal{M}')$.
Second, if $\mathcal{M} \models \Phi \cup \{\mathsf{E}((\varphi\,\mathcal{U}\,\psi) \wedge \chi)\}$ then there exists $\pi \in \mathsf{fullpaths}(\mathcal{M})$ such

that $\langle \mathcal{M}, \pi \rangle \models \Phi \cup \{(\varphi \mathcal{U} \psi) \wedge \chi\}$. or equivalently $\langle \mathcal{M}, \pi \rangle \models \Delta \cup \{\varphi \mathcal{U} \psi\}$. where $\Delta = \Phi \cup \{\chi\}$. Thus, as a particular case of the above proof, we obtain that there exists $\mathcal{M}'$ and $\pi' \in \mathsf{fullpaths}(\mathcal{M}')$ such that $\langle \mathcal{M}', \pi' \rangle \models \Phi \cup \{\chi\} \cup \{\psi \vee (\varphi \wedge \circ((\varphi \wedge \neg\Delta) \mathcal{U} (\psi \wedge \neg\Delta)))\}$. Henceforth,

$$\Phi \cup \{\mathsf{E}((\psi \vee (\varphi \wedge \circ((\varphi \wedge \neg\Delta) \mathcal{U} (\psi \wedge \neg\Delta)))) \wedge \chi)\}$$

is proved to be satisfiable.
Finally, let us assume that $\mathcal{M}_0 \models \Phi \cup \{\mathsf{A}((\varphi \mathcal{U} \psi) \vee \chi)\}$ holds for some $\mathsf{CTL}^\star$-structure $\mathcal{M}_0$. By Remark 1 we assume that χ is an state formula. In order to obtain a contradiction, let us suppose that

$$\mathcal{M} \not\models \Phi \cup \{\mathsf{A}(\psi \vee (\varphi \wedge \circ((\varphi \wedge \neg\Phi) \mathcal{U} (\psi \wedge \neg\Phi))) \vee \chi)\}$$

for all $\mathsf{CTL}^\star$-structure $\mathcal{M}$. Consequently, every $\mathcal{M}$ that is a model of Φ must also satisfy the negation of the just above A-formula or equivalently the formula

$$\mathsf{E}(\neg\psi \wedge \neg(\varphi \wedge \circ((\varphi \wedge \neg\Phi) \mathcal{U} (\psi \wedge \neg\Phi))) \wedge \neg\chi).$$

In particular, there must exists $\sigma_0 = s_0^0, s_1^0, \ldots \in \mathsf{fullpaths}(\mathcal{M}_0)$ such that

$$\langle \mathcal{M}_0, \sigma_0 \rangle \models \neg\psi \wedge \neg(\varphi \wedge \circ((\varphi \wedge \neg\Phi) \mathcal{U} (\psi \wedge \neg\Phi))) \wedge \neg\chi$$

By the initial hypothesis $\langle \mathcal{M}_0, \sigma_0 \rangle \models (\varphi \mathcal{U} \psi) \vee \chi$, then

$$\langle \mathcal{M}_0, \sigma_0 \rangle \models \{\neg\psi, \varphi, \circ(\varphi \mathcal{U} \psi), \neg\circ((\varphi \wedge \neg\Phi) \mathcal{U} (\psi \wedge \neg\Phi)), \neg\chi\}.$$

Therefore, there exists $j \geq 1$ such that

$$\langle \mathcal{M}_0, \mathsf{suffix}(\sigma_0, s_j^0) \rangle \models \Phi \cup \{\psi \vee (\varphi \wedge \circ((\varphi \wedge \neg\Phi) \mathcal{U} (\psi \wedge \neg\Phi))) \vee \chi)\}$$

The existence of such j can be ensured as follows. Let h be the smallest z such that $\langle \mathcal{M}_0, \mathsf{suffix}(\sigma_0, s_z^0) \rangle \models \psi$ which, since $\langle \mathcal{M}_0, \sigma_0 \rangle \models \circ(\varphi \mathcal{U} \psi)$, exists and is greater than zero. Now, if $\langle \mathcal{M}_0, \mathsf{suffix}(\sigma_0, s_h^0) \rangle \models \Phi$, then $j = h$. Otherwise, if $\langle \mathcal{M}_0, \mathsf{suffix}(\sigma_0, s_h^0) \rangle \not\models \Phi$ then $h > 1$ because $\langle \mathcal{M}_0, \sigma_0 \rangle \models \{\circ(\varphi \mathcal{U} \psi), \neg\circ((\varphi \wedge \neg\Phi) \mathcal{U} (\psi \wedge \neg\Phi))\}$. In this case j is assigned to be the greatest $x \in \{1, \ldots, h-1\}$ such that $\langle \mathcal{M}_0, \mathsf{suffix}(\sigma_0, s_x^0) \rangle \models \Phi$. Hence, in both cases, we have $\langle \mathcal{M}_0, \mathsf{suffix}(\sigma_0, s_j^0) \rangle \models \Phi$, so that also

$$\langle \mathcal{M}_0, \mathsf{suffix}(\sigma_0, s_j^0) \rangle \models \mathsf{E}(\neg\psi \wedge \neg(\varphi \wedge \circ((\varphi \wedge \neg\Phi) \mathcal{U} (\psi \wedge \neg\Phi))) \wedge \neg\chi).$$

and, therefore, there must exists at least a path $\sigma_1 = s_0^1, s_1^1, \ldots$ such that $s_0^1 = s_j^0$ and

$$\langle \mathcal{M}_0, \sigma_1 \rangle \models \neg\psi \wedge \neg(\varphi \wedge \circ((\varphi \wedge \neg\Phi) \mathcal{U} (\psi \wedge \neg\Phi))) \wedge \neg\chi.$$

Let λ_0 be the finite path $s_0^0, \ldots, s_{j-1}^0$ in $\mathsf{paths}(\mathcal{M}_0)$, then $\langle \mathcal{M}_0, \mathsf{suffix}(\sigma_0, s_\ell^0) \rangle \models \{\neg\psi, \varphi, \circ(\varphi \mathcal{U} \psi)\}$ for every $\ell \in \{0, \ldots, j-1\}$. Let $\mathcal{M}_1$ be the sub-structure of $\mathcal{M}_0$ whose root is s_j^0, then $\mathcal{M}_1$ is a $\mathsf{CTL}^\star$-structure such that $\sigma_1 \in \mathsf{fullpaths}(\mathcal{M}_1)$, $\pi_0 = \mathsf{suffix}(\sigma_0, s_j^0) \in \mathsf{fullpaths}(\mathcal{M}_1)$, and

$$\langle \mathcal{M}_1, \pi_0 \rangle \models \Phi \cup \{\psi \vee (\varphi \wedge \circ((\varphi \wedge \neg\Phi) \mathcal{U} (\psi \wedge \neg\Phi))) \vee \chi)\}$$

whereas $\langle \mathcal{M}_1, \sigma_1 \rangle \models \neg\psi \wedge \neg(\varphi \wedge \circ((\varphi \wedge \neg\Phi) \mathcal{U} (\psi \wedge \neg\Phi))) \wedge \neg\chi$. Therefore, the reasoning above applied to $\mathcal{M}_0$ can be identically applied to $\mathcal{M}_1$ allowing us to construct a finite path λ_1 such that $\sigma_1 = \lambda_1 \cdot \pi_1$ for some infinite π_1 and defining a sub-structure $\mathcal{M}_2$, that satisfies the same as $\mathcal{M}_1$, and so on this infinite process yields an infinite sequence of paths

$$\{\lambda_0 \cdot \pi_0, \quad \lambda_0 \cdot \lambda_1 \cdot \pi_1, \quad \lambda_0 \cdot \lambda_1 \cdot \lambda_2 \cdot \pi_2, \quad \cdots\} \subseteq \mathsf{fullpaths}(\mathcal{M}_0)$$

By the limit closure property, the infinite sequence $\lambda = \lambda_0 \cdot \lambda_1 \cdot \lambda_2 \cdot \lambda_3 \cdot \lambda_4 \cdot \ldots$ is a fullpath of $\mathcal{M}_0$. Additionally, $\langle \mathcal{M}_0, \lambda\rangle \not\models \varphi \,\mathcal{U}\, \psi$ and $\langle \mathcal{M}_0, \lambda\rangle \not\models \chi$. The former holds because $\langle \mathcal{M}_0, \mathsf{suffix}(\lambda, s_z)\rangle \models \neg\psi$ for all $z \geq 0$. The latter does because χ is a state formula, so that $\neg\chi$ is a state formula and $\langle \mathcal{M}_0, s_0^0\rangle \models \neg\chi$. As a consequence $\mathcal{M}_0 \not\models \Phi \cup \{\mathsf{A}((\varphi \,\mathcal{U}\, \psi) \vee \chi)\}$. Thus, we have proved by contradiction that there must exists $\mathcal{M}$ such that $\mathcal{M} \models \Phi \cup \{\mathsf{A}(\psi \vee (\varphi \wedge \circ((\varphi \wedge \neg\Phi)\,\mathcal{U}\,(\psi \wedge \neg\Phi))) \vee \chi)\}$.
The other context-dependent ϱ-rules are particular cases of $(\mathcal{U})_2$ rule. ∎

In addition, a correctness proof of a similar rule for CTL can be found in [2].

Theorem 27. (Soundness) *If there exists a closed expanded tableau for Φ then Φ is unsatisfiable.*

Proof. Let $\mathcal{T}_\Phi$ be a closed expanded tableau for Φ. Every bundle Y in $\mathcal{T}_\Phi$ is closed and contains at least one branch whose leaf is inconsistent. Consequently every node in $\mathcal{T}_\Phi$ is a failing node and in particular the root. Then, by the Lemma 26, Φ is unsatisfiable. ∎

5.2 Completeness

Here we prove that the tableau system BTM is complete by showing that we can associate with any open expanded bundle Y of the systematic tableau for Φ a cyclic CTL*-structure $\mathcal{G}_Y$ that yields a model of Φ.

We would like to remark that as soon as an open expanded bundle is detected, the algorithm $\mathcal{ST}$ could stop providing a model for the original set of formulae. Now, we associate with any open expanded bundle a CTL*-structure.

Definition 28. *Let $\mathcal{T}_\Phi$ be a systematic tableau for Φ. For any open expanded bundle Y in B, we define the* CTL**-structure $\mathcal{G}_Y = (S_{\mathcal{G}_Y}, R_{\mathcal{G}_Y}, V_{\mathcal{G}_Y})$ such that*

$S_{\mathcal{G}_Y} = \{s \in \mathsf{stages}(b) \mid b \in Y\}$
$R_{\mathcal{G}_Y} = \{(s, s') \in S_{\mathcal{G}_Y} \times S_{\mathcal{G}_Y} \mid (\mathsf{last}(s), \mathsf{first}(s')) \in R\}$[11]
$V_{\mathcal{G}_Y}(s) = \{\varepsilon \in L(s) \mid \varepsilon \in \mathsf{Prop}\}$. ∎

In the rest of this section we will assume that Y is an open expanded bundle of $\mathcal{T}_\Phi$ and that $\mathcal{G}_Y$ is the cyclic CTL*-structure associated to Y. Note that every branch $b = n_0, \ldots, n_k$ in Y is an expanded branch of $\mathcal{T}_\Phi$, hence every b is cyclic. The next lemma shows that each state of $\mathcal{G}_Y$ satisfies the set of formulae labeling all its nodes.

Lemma 29. *For every infinite $\pi \in \mathsf{paths}(\mathcal{G}_Y)$, if $\chi \in L(\mathsf{first}(\pi))$ then $\langle \mathcal{G}_Y, \pi\rangle \models \chi$.*

Proof. This fact is proved by a routine induction on χ. Most of the cases are easy consequences of the induction hypothesis and the fact that $L(s)$ is $\alpha\beta\varrho$-saturated for each in $s \in S_{\mathcal{G}_Y}$. Here we only detail the non-trivial cases.

Consider first the state formulae. The case $\chi = \varepsilon \in \mathsf{Prop}$ holds by definition of $\mathcal{G}_Y$. The case $\neg\varphi$ requires induction on φ but any of the generated cases holds, by the

[11] where R is the successor relation on $\mathcal{T}_\Phi$.

induction hypothesis, because $L(\mathsf{first}(\pi))$ is $\alpha\beta\varrho$-saturated. The same technique applies to $\varphi \vee \psi$ and $\varphi \wedge \psi$.

The case $\mathsf{A}\varphi$ is proved by induction on the inner formula φ. The induction here is based on a function rg that associates a natural number to each formula. We first define rg and then show that the lemma holds for the case $rg(\varphi) = 0$ and also that for each $\mathsf{A}\varphi$ such that $rg(\varphi) > 0$ there is a BTM-rule which applies to $\mathsf{A}\varphi$ and yields constituents whose range is less than $rg(\varphi)$. We define $rg(\varphi) = F_{\circ}(\varphi) + F_{\neg}(\varphi) + F_{\vee}(\varphi)$ where for any formula δ

$F_{\circ}(\delta)$ is the number of $\mathcal{U}, \Diamond, \Box$ in δ that are out of the scope of $\circ$
$F_{\neg}(\delta)$ is the number of $\neg, \vee, \wedge, \circ$ in δ that are out of the scope of $\neg$
$F_{\vee}(\delta)$ is the number of $\wedge, \circ$ in δ that are out of the scope of $\vee$

It is routine to check that every ϱ(-cd)-rule is decreasing with respect to rg.

Now, let $\pi = s_0, s_1, s_2, \ldots$ where each $L(s_i)$ is $\alpha\beta\varrho$-saturated.

On one hand, if $rg(\varphi) = 0$ then φ is a conjunction of elementary formulae of the form $\mathbf{F}$, $\mathbf{T}$, $\neg\mathbf{F}$, $\neg\mathbf{T}$, ε, $\neg\varepsilon$ and $\mathsf{Q}^b\circ\psi$. By rule $(\mathsf{A}^b\wedge)$, the case when φ has more than one conjunct is reduced to the case of one conjunct. If $\mathsf{AQ}^b\circ\psi \in L(s_0)$ then $\circ\psi \in L(s_0)$, by rule (Qdel). Thus, $\psi \in L(s_0)$ by (unnext^*). By induction hypothesis, $\langle \mathcal{G}_Y, s_1\rangle \models \psi$. So that $\langle \mathcal{G}_Y, \pi\rangle \models \mathsf{AQ}^b\circ\psi$. The other six cases of elementary formulae follows easily by (Qdel) and the definition $\mathcal{G}_Y$.

On the other hand, if $rg(\varphi) > 0$ then φ is a conjunction of formulae containing at least one conjunct φ' such that $rg(\varphi') > 0$. If $\mathsf{A}\varphi \in L(s_0)$, by $(\mathsf{A}^b\wedge)$, also $\mathsf{A}\varphi' \in L(s_0)$ for every conjunct φ' of φ. The conjuncts φ' such that $rg(\varphi') = 0$ are basic cases (showed above), thus $\langle \mathcal{G}_Y, s_0\rangle \models \mathsf{A}\varphi'$ holds for all them. For any φ' such that $rg(\varphi') > 0$, by the ϱ-rules and ϱ-cd-rules, $\mathsf{A}\varphi'' \in L(s_0)$ for some φ'' such that $rg(\varphi'') < rg(\varphi')$. Hence, by induction hypothesis, $\langle \mathcal{G}_Y, s_0\rangle \models \mathsf{A}\varphi''$ where φ'' depends on the form of the ϱ(-cd)-rule but, in every case, guarantee that $\langle \mathcal{G}_Y, s_0\rangle \models \mathsf{A}\varphi'$. Consequently, $\langle \mathcal{G}_Y, s_0\rangle \models \mathsf{A}\varphi$.

The case $\mathsf{E}\varphi$ is dual to $\mathsf{A}\varphi$ by defining $rg(\mathsf{E}\varphi) = F_{\circ}(\varphi) + F_{\neg}(\varphi) + F_{\wedge}(\varphi)$ and $F_{\wedge}(\varphi)$ is the number of $\vee, \circ$ in φ that are out of the scope of $\wedge$. The rule $(\mathsf{E}^b\vee)$ plays here the dual role to $(\mathsf{A}^b\vee)$. The definition of $\mathcal{G}_Y$ from a bundle Y enables the induction hypothesis when using $(\mathsf{E}^b\vee)$.

Regarding the path formulae, the cases $\neg\varphi$ (and its subcases, excepting $\neg\Box\psi$), $\varphi\vee\psi$, $\varphi \wedge \psi$, $\circ\varphi$ and $\Box\varphi$ are easy consequences of the induction hypothesis and the $\alpha\beta\varrho$-saturated property of stages. The case $\varphi\,\mathcal{U}\,\psi$ are straight consequences of Propositions 19 and 20, depending on whether $\varphi\,\mathcal{U}\,\psi$ is distinguished in stage $L(\mathsf{first}(\pi))$ or not. The case $\Diamond\varphi$ (and the subcase $\neg\Box\psi$ of $\neg\varphi$) are particular cases of the latter. ∎

Therefore $\mathcal{G}_Y$ is the desired model, i.e. $\mathcal{G}_Y \models \Phi$. Using this model construction we can easily prove (refutational) completeness of the tableau system BTM.

Theorem 30. (Refutational completeness) *If Φ is unsatisfiable then the systematic algorithm produces a closed tableau for Φ.* ∎

Given a satisfiable set Φ, by Lemma 24 the systematic algorithm produces a finite expanded tableau $\mathcal{T}_\Phi$ for Φ. If $\mathcal{T}_\Phi$ is closed then, by Theorem 27, Φ is unsatisfiable. Since this contradicts that Φ is satisfiable, $\mathcal{T}_\Phi$ must be necessarily an open expanded tableau. Consequently, we can ensure completeness.

Theorem 31. *If Φ is satisfiable then there exists a (finite) open expanded tableau for Φ.*

■

6 Conclusions

We have defined a tableau system for the full computation tree logic $\mathsf{CTL}^\star$ and the algorithm for the systematic construction of the tableau construction for any set of $\mathsf{CTL}^\star$ formulae and established the correctness of these developments. We are not aware of any other one-pass tableau construction for $\mathsf{CTL}^\star$. For this expressive logic, which has huge applications, the repository of deductive methods is currently restricted to only a few, quite recently defined, formalisms, both due to Reynolds, namely the axiomatics [9] and the tableau [10]. However, we should note that both of these developments are quite complex and we can see an obvious benefit of our system, first of all, in the transparency of the rules, their intuitive clearness, in providing a method that efficiently limits the length of branches –a problem that stayed open in [10]– and, finally, in the algorithm of the systematic tableau construction. Of course, the algorithm itself may be subject to a further investigation into its refinement and this would form part of our future work as well as its implementation.

The results presented in this paper open very good prospect for the development of a corresponding dual cut-free sequent calculus for $\mathsf{CTL}^\star$ in line with [7]. We can also see a basis for defining an invariant-free resolution method for the full computation tree logic extending the results in [6] and complementing the existing clausal resolution method for CTL-type branching time logics, initially set up in [1]. Finally, noting that in this paper we have not mentioned anything about the complexity issues, relevant complexity study applied to the tableau algorithm and its refinements will also constitute part of our future work.

References

1. A. Bolotov and M. Fisher. A clausal resolution method for CTL branching-time temporal logic. *J. Exp. Theor. Artif. Intell.*, 11(1):77–93, 1999.
2. K. Brünnler and M. Lange. Cut-free sequent systems for temporal logic. *Journal of Logic and Algebraic Programming*, 76(2):216–225, 2008.
3. E. M. Clarke and E. A. Emerson. Design and synthesis of synchronisation skeletons using branching time temporal logic. In *Logic of Programs. Proceedings of Workshop*, volume 131 of Lecture Notes in Computer Science, pages 52–71. Springer-Verlag, 1981.
4. E. A. Emerson and A. P. Sistla. Deciding full branching time logic. In *STOC 1984, Proceedings of*, pages 14–24, 1984.
5. E. Allen Emerson. Temporal and modal logic. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 995–1072. 1990.
6. J. Gaintzarain, M. Hermo, P. Lucio, M. Navarro, and F. Orejas. A cut-free and invariant-free sequent calculus for PLTL. In J. Duparc and T. A. Henzinger, editors, *Computer Science Logic, 21st International Workshop, CSL 2007, 16th Annual Conference of the EACSL, Lausanne, Switzerland, September 11-15, 2007, Proceedings*, volume 4646 of *Lecture Notes in Computer Science*, pages 481–495. Springer, 2007.

7. J. Gaintzarain, M. Hermo, P. Lucio, M. Navarro, and F. Orejas. Dual systems of tableaux and sequents for PLTL. *Journal of Logic and Algebraic Programming*, 78(8):701–722, 2009.
8. R. Gore. *Tableau methods for modal and temporal logics*, pages 297–396. Kluwer Academic Publishers, 1999.
9. M. Reynolds. An axiomatization of full computation tree logic. *J. Symb. Log.*, 66(3):1011–1057, 2001.
10. M. Reynolds. A tableau for CTL*. In Ana Cavalcanti and Dennis Dams, editors, *FM 2009: Formal Methods, Second World Congress, Eindhoven, The Netherlands, November 2-6, 2009. Proceedings*, volume 5850 of *Lecture Notes in Computer Science*, pages 403–418. Springer, 2009.
11. M. Reynolds and C. Dixon. Theorem-proving for discrete temporal logic. In *Handbook of Temporal Reasoning in Artificial Intelligence*, pages 279–314, 2005.