Branching-Time Logic $\mathsf{ECTL}^{\#}$ and its tree-style one-pass tableau: Extending Fairness Expressibility of ECTL^{+} [⋆]

TECHNICAL REPORT

JANUARY 20, 2020

Alexander Bolotov[1], Montserrat Hermo[2], and Paqui Lucio[2]

[1] University of Westminster, W1W 6UW, London, UK.
[2] University of the Basque Country, 20018-San Sebastián, Spain.

Abstract. Temporal logic has become essential for various areas in computer science, most notably for the specification and verification of hardware and software systems. For the specification purposes rich temporal languages are required that, in particular, can express fairness constraints. For linear-time logics which deal with fairness in the linear-time setting, one-pass and two-pass tableau methods have been developed. In the repository of the CTL-type branching-time setting, the well-known logics ECTL and ECTL^{+} were developed to explicitly deal with fairness. However, due to the syntactical restrictions, these logics can only express restricted versions of fairness. The logic $\mathsf{CTL}^{\star}$, often considered as 'the full branching-time logic' overcomes these restrictions on expressing fairness. However, $\mathsf{CTL}^{\star}$ is extremely challenging for the application of verification techniques, and the tableau technique, in particular. For example, there is no one-pass tableau construction for $\mathsf{CTL}^{\star}$, while one-pass tableau has an additional benefit enabling the formulation of dual sequent calculi that are often treated as more 'natural' being more friendly for human understanding. These two considerations lead to the following problem - are there logics that have richer expressiveness than ECTL^{+}, allowing the formulation of a new range of fairness constraints with 'until' operator, yet 'simpler' than $\mathsf{CTL}^{\star}$, and for which a one-pass tableau can be developed? Here we give a positive answer to this question, introducing a sub-logic of $\mathsf{CTL}^{\star}$ called $\mathsf{ECTL}^{\#}$, its tree-style one-pass tableau, and an algorithm for obtaining a systematic tableau, for any given admissible branching-time formulae. We prove the termination, soundness and completeness of the method. As tree-shaped one-pass tableaux are well suited for the automation and are amenable for the implementation and for the formulation of sequent calculi. Our results also open a prospect of relevant developments of the automation and implementation of the tableau method for $\mathsf{ECTL}^{\#}$, and of a dual sequent calculi.

1 Introduction

Temporal logic has become essential for the specification and verification of hardware and software systems. For the specification of the reactive and distributed systems, or, most recently, autonomous systems, the modelling of the possibilities 'branching' into the future is essential. Among important properties of these systems, so called fairness properties are important. In the standard formalisation of fairness, operators $\Diamond$ (eventually) and $\Box$ (always) have been used: $\mathsf{A}\Diamond\Box p$ – 'p' is true along all computation paths except possibly their finite initial interval, where 'A' is 'for all paths' quantifier, and $\mathsf{E}\Box\Diamond p$ – 'p' is true along a computation path at infinitely many states, where 'E' stands for 'there exists a path' quantifier. Branching-time logics (BTL) here give us an appropriate reasoning framework, where the most used class of formalisms are 'CTL' (Computation Tree Logic) type logics. CTL itself requires every temporal operator to be preceded by a path quantifier, thus, cannot express fairness. ECTL (Extended CTL) [8] enables simple fairness constraints but not their Boolean combinations. ECTL^{+} [9] further extends the expressiveness of ECTL allowing Boolean combinations of temporal operators and ECTL fairness constraints (but not permitting their nesting). The logic $\mathsf{CTL}^{\star}$, often considered as 'the full branching-time logic' overcomes these restrictions on expressing fairness. However, $\mathsf{CTL}^{\star}$ is extremely challenging for the application of any known technique of automated reasoning. Note that, unlike fair CTL [6] which, in tackling

⋆ This work has been partially supported by Spanish Projects TIN2013-46181-C2-2-R and TIN2017-86727-C2-2-R, and by the University of the Basque Country under Project LoRea GIU15/30.

fairness, changes the underlying trees to those with 'fair paths' only, ECTL and ECTL^+ do not impose these changes.

From another perspective, the literature on fairness constraints, even in the linear-time setting, lacks the analysis of their formulation with the $\mathcal{U}$ ('until') operator. To the best of our knowledge, there are only a few research papers that raise or discuss the problem. For example, [15], introduces the logic LCTL, providing an extension of liveness constraints by the "until" operator. However, LCTL belongs to 'Fair CTL-type' logics [10]. 'Generalised liveness assumptions, which allow to express that the conclusion $f_2 \mathcal{U} f_3$ of a liveness assumption $\Box(f_1 \Rightarrow (f_2 \mathcal{U} f_3))$ has to be satisfied' are addressed in [2]. The $\mathcal{U}$ operator in the formulation of the fairness can also be found in [25] which considers the sequential composition of processes, providing the following example - the composition of processes P_1 and P_2 'behaves as $P1$ until its termination and then behaves as $P2$'. Finally, [16] utilises restricted linear-time fairness constraints with $\mathcal{U}$ in the linear-time setting. We are not aware of any other analysis of fairness constraints in branching-time setting using the $\mathcal{U}$ operator and without restricting the underlying logic to be interpreted over the 'fair' paths. We bridge this gap, presenting the logic $\mathsf{ECTL}^\#$ (we use $\#$ to indicate some restrictions on concatenations of the modalities and their Boolean combinations). It is weaker than $\mathsf{CTL}^\star$ but extends ECTL^+ by allowing the combinations $\Box(A \mathcal{U} B)$ or $A \mathcal{U} \Box B$, referred to as modalities $\Box\mathcal{U}$ and $\mathcal{U}\Box$. This enables the formulation of stronger fairness constraints in the branching-time setting. The fairness constraint $\mathsf{A}(p \mathcal{U} \Box q)$ reads as 'invariant q is true along all paths of the computation except possibly their finite initial interval, where p is true'. For example, the following property specifies that whenever the user of an account is requested to change the current password, either it is changed to a fresh one, or the account is deactivated:

$$\mathsf{A}((P_n^w \mathcal{U} \Box(R_n \Rightarrow \mathsf{A}\Box(P_n^{w'} \Rightarrow w \neq w')) \vee ((L_n \wedge P_n^w) \mathcal{U} \Box((R_n \wedge (\neg P_n^{w'} \vee w' = w)) \Rightarrow \neg L_n))) \quad (1)$$

where P_n^w ($P_n^{w'}$) stands for the account n has an associated password w (w'); L_n stands for the account n is live, and R_n means the account number n is requested to change the password, Note that formula (1) represents one of the difficult cases of $\mathsf{ECTL}^\#$ structures - an A-disjunctive formula, see §2.

$\mathcal{B}(\mathcal{U}, \circ)$ (CTL) extensions	$\mathsf{E}(\Box\Diamond q)$	$\mathsf{E}(\Box\Diamond q \wedge \Box\Diamond r)$	$\mathsf{A}((p \mathcal{U} \Box q) \vee (s \mathcal{U} \Box \neg r))$	$\mathsf{A}\Diamond(\circ p \wedge \mathsf{E}\circ\neg p)$
$\mathcal{B}(\mathcal{U}, \circ, \Box\Diamond)$ (ECTL)	$\surd$	X	X	X
$\mathcal{B}^+(\mathcal{U}, \circ, \Box\Diamond)$ (ECTL^+)	$\surd$	$\surd$	X	X
$\boldsymbol{\mathcal{B}^+(\mathcal{U}, \circ, \mathcal{U}\Box)}$ ($\mathsf{ECTL}^\#$)	$\surd$	$\surd$	$\surd$	X
$\mathcal{B}^\star(\mathcal{U}, \circ)$ ($\mathsf{CTL}^\star$)	$\surd$	$\surd$	$\surd$	$\surd$

Fig. 1: Classification of CTL-type logics and their expressiveness

Figure 1, which utilises another temporal operator - $\circ$ - 'at the next moment of time', places our logic in the hierarchy of BTL representing their expressiveness: logics are classified by using '$\mathcal{B}$' for 'Branching', followed by the set of only allowed modalities as parameters; $\mathcal{B}^+$ indicates admissible Boolean combinations of the modalities and $\mathcal{B}^\star$ reflects 'no restrictions' in either concatenations of the modalities or Boolean combinations between them.[3] Thus, $\mathcal{B}(\mathcal{U}, \circ)$ denotes the logic CTL. In this hierarchy $\mathsf{ECTL}^\#$ is $\mathcal{B}^+(\mathcal{U}, \circ, \mathcal{U}\Box)$.

We present a tree-style one-pass tableau for $\mathsf{ECTL}^\#$ continuing the analogous developments in linear-time case [4, 13] and for CTL [4]. An indicative feature of this approach is a context-based tableau technique. Context-based tableaux have dual sequent calculi due to their handling of eventualities exclusively by using logical rules. To the best of our knowledge, BTL more expressive than CTL have not enjoyed the context-based tableau though other kinds of tableaux exist for these logics. There is a single-pass tableau for CTL that carries out an 'on the fly' eventualities checking (non-logical mechanism) following the Schwendimann's approach [1]. For $\mathsf{CTL}^\star$, which definitely is a super-logic of $\mathsf{ECTL}^\#$, different other kinds of tableau-style methods exist, remarkably [12, 21, 18, 22, 23]. Since $\mathsf{CTL}^\star$ is much more expressive than $\mathsf{ECTL}^\#$, such methods often utilise additional mechanisms (non only inference rules) to

[3] This notation goes back to [7], here we use its nice tuning by Nicolas Markey in [19]. In the last column we use a short $\mathsf{CTL}^\star$ formula $\mathsf{A}\Diamond(\circ p \wedge \mathsf{E}\circ\neg p)$, not expressible by weaker logics. We found this formula indicative for $\mathsf{CTL}^\star$ as its validity is directly linked to the limit closure property [7].

control loops, which are, for example, automata-theoretic-based mechanism [22]. This brings extra complexity, which is justified to handle the $\mathsf{CTL}^\star$ expressivity. However, simpler proofs could be obtained for a weaker logic such as $\mathsf{ECTL}^{\#}$. There are also extensions of the tableau methods to super-logics of $\mathsf{CTL}^\star$. For example, [5] introduces a two-pass tableau method for a logic that is a multiagent extension of $\mathsf{CTL}^\star$. Tree-style one-pass tableaux (without additional procedures for checking meta-logical properties) have dual (cut-free) sequent calculi, see [13], enabling the construction of human-understandable proofs. In addition, these tableaux are well suited for the automation and are amenable for the implementation.[4] Our tableau is effectively an AND-OR tree where nodes are labelled by sets of state (see the definitions in §2) formulae. There are difficult cases of $\mathsf{ECTL}^{\#}$ formulae that appear due to the enriched syntax: disjunctions of formulae in the scope of the A quantifier and conjunctions of formulae in the scope of the E quantifier. To tackle these cases, in addition to $\alpha - \beta$ rules, that are standard to the tableaux, we define novel β^+-rules which use the context to force the eventualities to be fulfilled as soon as possible.

Outline of the paper. The rest of this paper, an extended version of [3], includes more examples, explanations, and detailed proofs of the results. It commences with §2 where we describe $\mathsf{ECTL}^{\#}$ as a sublogic of $\mathsf{CTL}^\star$. The formulation of the tableau method is given in §3, where we define and explain tableau rules. A *systematic* tableau construction and relevant examples are introduced in §4. More examples and a set of derived rules are presented in §5. The soundness and completeness of our tableau method are proved in §6 and in §7, respectively; for the latter, we prove the refutational completeness and termination of the presented method. In §8, we analise the time complexity of the tableau method. Finally, in §9 we draw the conclusions and prospects of future work that the presented results open.

[4] An excellent survey of the seminal tableau techniques for temporal logics can be found in [14].

2 The logic $\mathsf{ECTL}^{\#}$

As $\mathsf{ECTL}^{\#}$ is a sublogic of $\mathsf{CTL}^{\star}$ we first recall $\mathsf{CTL}^{\star}$ syntax and semantics.

Definition 1 (Syntax of $\mathsf{CTL}^{\star}$). *Given Prop is a fixed set of propositions, and $p \in \mathsf{Prop}$, we define sets of state (σ) and path (π) $\mathsf{CTL}^{\star}$ formulae over Prop as follows: $\sigma ::= \mathbf{T} \mid p \mid \sigma_1 \wedge \sigma_2 \mid \neg\sigma \mid \mathsf{E}\pi$ and $\pi ::= \sigma \mid \pi_1 \wedge \pi_2 \mid \neg\pi \mid \circ\pi \mid \pi\,\mathcal{U}\,\pi \mid \Box\pi$.* ■

In $\mathsf{CTL}^{\star}$, and all BTL logics, well formed formulae are state formulae.

Definition 2 (Labelled Kripke structure). *A Kripke structure, $\mathcal{K}$, is a triple (S, R, L) where $S \neq \emptyset$ is a set of states, $R \subseteq S \times S$ is a total binary relation, called the transition relation, and $L : S \to 2^{\mathsf{Prop}}$ is a labelling function.* ■

A *fullpath* x through a Kripke structure $\mathcal{K}$ is an infinite sequence of states $s_0, s_1, \ldots$ such that $(s_i, s_{i+1}) \in R$, for every $i \geq 0$. Let '$\mathsf{fullpaths}(\mathcal{K})$' be the set of all fullpaths in $\mathcal{K}$. Given a fullpath $x = s_0, s_1, \ldots, s_k, \ldots$ $(k \geq 0)$, we denote its state s_k by $x(k)$, its finite prefix by the sequence $x^{\leq k} = s_0, s_1, \ldots, s_k$ and the suffix path $x^{\geq k} = s_k, s_{k+1}, \ldots$. When a fullpath x is given, instead of $x(k)$ we will often write k, referring to k as 'a state index of x'. If x is a fullpath and y is a path such that $y(0) = x(k)$, for some $k > 0$, then the juxtaposition $x^{\leq k}y$ is a fullpath. Our Kripke structures are labelled directed graphs that correspond to Emerson's R-generable structures, i.e. the transition relation R is suffix, fusion and limit closed [7]. For any $\mathcal{K}$, any $x \in \mathsf{fullpaths}(\mathcal{K})$ and any natural number i, the notation $\mathcal{K} \upharpoonright x(i)$ denotes a Kripke structure with the set of states of $\mathcal{K}$ restricted to those that are R-reachable from $x(i)$.

Definition 3. *Given the structure $\mathcal{K} = (S, R, L)$, the relation $\models$, which evaluates path formulae in a given path x and state formulae at the state index i of the given path x, is defined bellow:*

$\mathcal{K}, x, i \models \mathbf{T}$	*and*	$\mathcal{K}, x, i \models p$ *iff* $p \in L(x(i))$.
$\mathcal{K}, x, i \models \neg\sigma$	*iff*	$\mathcal{K}, x, i \models \sigma$ *does not hold.*
$\mathcal{K}, x, i \models \sigma_1 \wedge \sigma_2$	*iff*	$\mathcal{K}, x, i \models \sigma_1$ *and* $\mathcal{K}, x, i \models \sigma_2$.
$\mathcal{K}, x, i \models \mathsf{E}\pi$	*iff*	*there exists a path* $y \in \mathsf{fullpaths}(\mathcal{K} \upharpoonright x(i))$ *such that* $\mathcal{K}, y \models \pi$.
$\mathcal{K}, x \models \circ\pi$	*iff*	$\mathcal{K}, x^{\geq 1} \models \pi$.
$\mathcal{K}, x \models \neg\pi$	*iff*	$\mathcal{K}, x \models \pi$ *does not hold.*
$\mathcal{K}, x \models \pi_1 \wedge \pi_2$	*iff*	$\mathcal{K}, x \models \pi_1$ *and* $\mathcal{K}, x \models \pi_2$.
$\mathcal{K}, x \models \pi_1\,\mathcal{U}\,\pi_2$	*iff*	*there exists* $k \geq i$ *such that* $\mathcal{K}, x^{\geq k} \models \pi_2$ *and* $\mathcal{K}, x^{\geq j} \models \pi_1$ *for all* $0 \leq j \leq k-1$.
$\mathcal{K}, x \models \Box\pi$	*iff*	$\mathcal{K}, x^{\geq j} \models \pi$ *for all* $j \geq 0$.

In addition, for any set Σ of state formulae, $\mathcal{K}, x, i \models \Sigma$ iff $\mathcal{K}, x, i \models \sigma$, for all $\sigma \in \Sigma$. ■

Many other usual operators can be derived from those introduced, in particular, the 'falsehood' constant $\mathbf{F} \equiv \neg\mathbf{T}$, and the disjunction operator $\varphi_1 \vee \varphi_2 \equiv \neg(\neg\varphi_1 \wedge \neg\varphi_2)$, as well as the temporal operator $\Diamond\pi \equiv \mathbf{T}\,\mathcal{U}\,\pi$ and the universal path quantifier $\mathsf{A}\pi \equiv \neg\mathsf{E}\neg\pi$. It is also known that $\Box\pi \equiv \neg\Diamond\neg\pi$ but, for technical convenience, we define it as a primitive operator. Let us recall some meta-logical concepts that are essential for the paper.

Definition 4 (Syntactically Consistent Set of Formulae). *A set Σ of state formulae σ is syntactically consistent abbreviated as $\Sigma_\top$ if $\mathbf{F} \notin \Sigma$ and $\{\sigma, \neg\sigma\} \not\subseteq \Sigma$ for any σ; otherwise, Σ is inconsistent denoted as $\Sigma_\perp$.* ■

Definition 5 (Satisfiability). *For a set of state formulae Σ, the set of its models, $\mathsf{Mod}(\Sigma)$, is formed by all triples $(\mathcal{K}, x, i)$ such that $\mathcal{K}, x, i \models \Sigma$. Σ is satisfiable $(\mathsf{Sat}(\Sigma))$ if $\mathsf{Mod}(\Sigma) \neq \emptyset$, otherwise Σ is unsatisfiable $(\mathsf{UnSat}(\Sigma))$.* ■

If $\mathsf{Mod}(\Sigma) = \mathsf{Mod}(\Sigma')$ then Σ and Σ' are equivalent denoted as $\Sigma \equiv \Sigma'$. For a set of state formulae Σ, if for any fullpath $x \in \mathsf{fullpaths}(\mathcal{K})$, we have $\mathcal{K}, x, 0 \models \Sigma$, then we simply write $\mathcal{K} \models \Sigma$.

Definition 6 (Cyclic Sequence, Cyclic Path and Cyclic Kripke structure). *Let z be a finite sequence of states $z = s_0, s_1, \ldots, s_j$ such that, for every $0 \leq k < j$, $(s_k, s_{k+1}) \in R$. Then, z is cyclic iff there exists s_i, $0 \leq i \leq j$ such that $(s_j, s_i) \in R$. Let z be a finite cyclic sequence, the subsequence $s_i, \ldots, s_j$ of z is called a loop and s_i is called the cycling element. We denote the loop as $\langle s_i, \ldots, s_j\rangle^\omega$. A cyclic path over z is an infinite sequence $\mathsf{path}(z) = s_0, s_1, \ldots, s_{i-1}\langle s_i, s_{i+1}, \ldots, s_j\rangle^\omega$.*[5] *A Kripke structure $\mathcal{K}$ is cyclic if every fullpath is a cyclic path over a cyclic sequence of states.* ∎

The fact that $\mathsf{CTL}^\star$ satisfiability can be reduced to the emptiness problem for automata on infinite trees (see [24, 17]), ensures that the (non-empty) collection of models of a given satisfiable $\mathsf{CTL}^\star$ formula can be obtained by infinitely unwinding (in any possible way) a finite graph. Hence, for any $\mathsf{CTL}^\star$ formula ϕ, such that $\mathsf{Mod}(\phi) \neq \emptyset$, there always exists a model $\mathcal{K} \in \mathsf{Mod}(\phi)$ such that $\mathcal{K}$ is cyclic. Therefore, when speaking about the satisfiability in $\mathsf{CTL}^\star$ (hence $\mathsf{ECTL}^\#$) we can consider *cyclic* Kripke structures.

Proposing a new logic, $\mathsf{ECTL}^\#$, we aim at defining a sublogic of $\mathsf{CTL}^\star$ that extends the ECTL^+ formulae $\Box\Diamond\sigma$ and $\Diamond\Box\sigma$ (where σ means state formula), respectively, to $\Box(\sigma\,\mathcal{U}\,\sigma)$ and $\sigma\,\mathcal{U}\,\Box\sigma$.

Definition 7 (Syntax of $\mathsf{ECTL}^\#$). *The set of $\mathsf{ECTL}^\#$ formulae, over a fixed set of propositions Prop, are formed according to the following restriction of the $\mathsf{CTL}^\star$ grammar in Definition 1 for path formulae (state formulae are the same):* $\pi ::= \sigma \mid \pi_1 \wedge \pi_2 \mid \neg\pi \mid \circ\sigma \mid \sigma\,\mathcal{U}\,\sigma \mid \Box\sigma \mid \sigma\,\mathcal{U}\,(\Box\sigma) \mid \Box(\sigma\,\mathcal{U}\,\sigma)$.

Note that the nesting of pure path formulae, totally unrestricted in $\mathsf{CTL}^\star$, is now restricted by the grammar cases: $\circ\sigma \mid \sigma\,\mathcal{U}\,\sigma \mid \Box\sigma \mid \sigma\,\mathcal{U}\,\Box\sigma \mid \Box(\sigma\,\mathcal{U}\,\sigma)$. In particular, $a\,\mathcal{U}\,\Box(b \wedge \Box c)$ (where $a, b, c \in \mathsf{Prop}$) is not an $\mathsf{ECTL}^\#$ formula because $b \wedge \Box c$ is directly in the scope of the $\Box$ but is not a state formula. For technical convenience, we assume that the tableau construction applies to the formulae in *negation normal form* (shortly, nnf). Therefore, we introduce here a grammar for the set of $\mathsf{ECTL}^\#$ formulae that is closed under negation and requires the negation to apply to atomic propositions (instead of state and path formulae).

Definition 8 (Syntax of $\mathsf{ECTL}^\#$ in nnf). *Let Prop be a fixed set of propositions, let $\rho \in \mathsf{Prop}$ and let $Lit ::= \mathbf{F} \mid \mathbf{T} \mid \rho \mid \neg\rho$, be the set of literals. The set $\mathcal{F}_{\mathsf{Prop}}$ of $\mathsf{ECTL}^\#$ formulae in nnf (over Prop) is given by the grammar:*

$\sigma ::= Lit \mid \sigma_1 \wedge \sigma_2 \mid \sigma_1 \vee \sigma_2 \mid \mathsf{E}\pi \mid \mathsf{A}\pi$

$\pi ::= \pi_1 \wedge \pi_2 \mid \pi_1 \vee \pi_2 \mid \circ\sigma \mid \sigma\,\mathcal{U}\,(\Box\sigma) \mid \Box(\sigma\,\mathcal{U}\,\sigma) \mid \Box(\sigma \vee \Box\sigma) \mid \sigma\,\mathcal{U}\,(\sigma \wedge \Diamond\sigma)$

where σ means a state formula, π means a path formula, and $\Diamond\sigma$ abbreviates $\mathbf{T}\,\mathcal{U}\,\sigma$. ∎

The modified grammar is obtained by extending the state formulae grammar by $\mathsf{A}\pi$-formulae and the path formulae grammar by $\Box(\sigma \vee \Box\sigma)$ and $\sigma\,\mathcal{U}\,(\sigma \wedge \Diamond\sigma)$. Cases $\sigma\,\mathcal{U}\,\sigma$ and $\Box\sigma$ are omitted because they respectively abbreviate $\sigma\,\mathcal{U}\,(\sigma \wedge \Diamond\mathbf{T})$ and $\Box(\sigma \vee \Box\mathbf{F})$. Note that, for $a, b, c \in \mathsf{Prop}$, the formula $\Box(a \vee \Box(b \vee \Box c))$ is not in $\mathcal{F}_{\mathsf{Prop}}$ because $b \vee \Box c$ is not a state formula. The following proposition ensures that the set $\mathcal{F}_{\mathsf{Prop}}$ is closed under negation.

Proposition 9 (Closure under Negation). *For any $\varphi \in \mathcal{F}_{\mathsf{Prop}}$, we also have $\mathsf{nnf}(\neg\varphi) \in \mathcal{F}_{\mathsf{Prop}}$. Moreover, the negation of a state (resp. path) formula is a state (resp. path) formula.*

Proof. By structural induction on the formulae, using the following equivalences (and well known classical ones):

1. $\neg\mathsf{A}\varphi \equiv \mathsf{E}\neg\varphi$ 5. $\neg\Box(\varphi_1\,\mathcal{U}\,\varphi_2) \equiv \Diamond\Box\neg\varphi_2 \vee \Diamond((\neg\varphi_1) \wedge (\neg\varphi_2))$
2. $\neg\mathsf{E}\varphi \equiv \mathsf{A}\neg\varphi$ 6. $\neg(\varphi_1\,\mathcal{U}\,\Box\varphi_2) \equiv (\Box\Diamond\neg\varphi_2) \vee \Diamond(\neg\varphi_1 \wedge \Diamond\neg\varphi_2)$
3. $\neg\circ\varphi \equiv \circ\neg\varphi$ 7. $\neg(\varphi_1\,\mathcal{U}\,(\varphi_2 \wedge \Diamond\varphi_3)) \equiv \Box(\neg\varphi_2 \vee \Box\neg\varphi_3) \vee ((\neg\varphi_2)\,\mathcal{U}\,(\neg\varphi_1 \wedge \neg\varphi_2))$
4. $\neg\Box\varphi \equiv \Diamond\neg\varphi$

Equivalences 1-5 are very well known (e.g. [7]); the validity of 6 and 7 is easily established. It is also easy to see that 7, when φ_3 is $\mathbf{T}$, is reduced to the known equivalence $\neg(\varphi_1\,\mathcal{U}\,\varphi_2) \equiv (\Box\neg\varphi_2) \vee (\neg\varphi_2\,\mathcal{U}\,(\neg\varphi_1 \wedge \neg\varphi_2))$ ∎

For simplicity, we will write $\neg\varphi$ instead of $\mathsf{nnf}(\neg\varphi)$. Thus, $\neg\mathsf{A}(p\,\mathcal{U}\,\Box q)$ represents $(\mathsf{E}\Box\Diamond\neg q) \vee \mathsf{E}\Diamond(\neg p \wedge \Diamond\neg q))$. Also, for a finite set $\Delta = \{\varphi_1, \ldots, \varphi_n\}$, we let $\mathsf{nnf}(\neg\bigwedge_{i=1}^{n}\varphi_i) = \neg\Delta$.

[5] Cyclic paths are also known as *ultimately periodic* paths.

Type of a difficult case	A-disjunctive formula	E-conjunctive formula
Example	$A(\circ q \ \vee\ \Box r)$	$E(\circ r \ \wedge\ q\,\mathcal{U}\,\Box\neg p)$
Our representation	$A(\circ q, \Box r)$	$E(\circ r, q\,\mathcal{U}\,\Box\neg p)$

Fig. 2: Difficult cases of temporal operators in the scope of path quantifiers

For $\mathsf{ECTL}^{\#}$, we identify the following difficult cases of the nesting and Boolean combinations of temporal operators in the scope of path quantifiers: A-disjunctive formula – disjunctions of temporal operators in the scope of A and E-conjunctive formula – conjunctions of temporal operators in the scope of E. For convenience, we will, respectively, write $\mathsf{A}(\pi_1, \ldots, \pi_n)$ and $\mathsf{E}(\pi_1, \ldots, \pi_n)$, where $n \geq 1$, and if "," is in the scope of A it means $\vee$ while being in the scope of E it means $\wedge$. Formulae serving as relevant examples in Figure 2 will be used to illustrate tableau, in Figure 6. Note that any A-formula (E-formula) σ can be transformed into an equivalent Boolean combination of A-disjunctive formulae $\mathsf{A}(\pi_1, \ldots, \pi_n)$ (E-conjunctive formulae $\mathsf{E}(\pi_1, \ldots, \pi_n)$), such that every π_i $(1 \leq i \leq n)$ is of one of the following: $\circ\sigma$, $\sigma\,\mathcal{U}\,(\sigma \wedge \Diamond\sigma)$, $\sigma\,\mathcal{U}\,\Box\sigma$, $\Box(\sigma \vee \Box\sigma)$, and $\Box(\sigma\,\mathcal{U}\,\sigma)$, and σ stands for a state formula. For example, $\mathsf{A}(((\circ q) \wedge (\Box\mathsf{E}\circ r)) \vee \circ p)$ is equivalent to $\mathsf{A}(\circ q) \wedge \mathsf{A}(\Box\mathsf{E}\circ r, \circ p)$; and $\mathsf{E}(((\circ\mathsf{A}\circ r) \vee (q\,\mathcal{U}\,\Box\mathsf{E}\neg p)) \wedge \circ q)$ is equivalent to $\mathsf{E}(\circ\mathsf{A}\circ r) \vee \mathsf{E}(q\,\mathcal{U}\,\Box\mathsf{E}\neg p, \circ q)$. In what follows, Q abbreviates either of the path quantifiers. For a set of path formulae $\Pi = \{\pi_1, \ldots, \pi_n\}$, we write $\mathsf{Q}\Pi$ to denote $\mathsf{Q}(\pi_1, \ldots, \pi_n)$, and $\mathsf{Q}\circ\Pi$ to denote $\mathsf{Q}(\circ\pi_1, \ldots, \circ\pi_n)$. If Φ is an empty set of formulae it means $\mathbf{T}$ when Φ occurs in a conjunctive expression, and $\mathbf{F}$ in a disjunctive expression. In particular, when Π is $\emptyset$ then $\mathsf{A}\Pi$ is $\mathbf{F}$ and $\mathsf{E}\Pi$ is $\mathbf{T}$. We write Σ, σ to represent the set $\Sigma \cup \{\sigma\}$. We consider that every formula $\sigma \in \mathcal{F}_{\mathsf{Prop}}$ is given in its equivalent 'negation normal form', $\mathsf{nnf}(\sigma)$.

3 The Tableaux Method

In this section, we introduce a set of tableau rules and the method to apply them to construct a tableau.

	α	S_α
$(\wedge)$	$\sigma_1 \wedge \sigma_2$	$\{\sigma_1, \sigma_2\}$
$(\mathsf{E}\sigma)$	$\mathsf{E}(\sigma_1, \ldots, \sigma_n, \Pi)$	$\{\sigma_1, \ldots, \sigma_n, \mathsf{E}\Pi\}$
$(\mathsf{E}\Box\mathcal{U})$	$\mathsf{E}(\Box(\sigma_1 \mathcal{U} \sigma_2), \Pi)$	$\{\mathsf{E}(\sigma_1 \mathcal{U} \sigma_2, \circ\Box(\sigma_1 \mathcal{U} \sigma_2), \Pi)\}$
$(\mathsf{A}\Box\mathcal{U})$	$\mathsf{A}(\Box(\sigma_1 \mathcal{U} \sigma_2), \Pi)$	$\{\mathsf{A}(\sigma_1 \mathcal{U} \sigma_2, \Pi), \mathsf{A}(\circ\Box(\sigma_1 \mathcal{U} \sigma_2), \Pi)\}$

Fig. 3: ALPHA RULES. (Notation: σ, σ_i stand for state formulae and Π is a set of path formulae, possibly empty.)

β_Rule	β	k	$S_{\beta_i}(1 \leq i \leq k)$
$(\vee)$	$\sigma_1 \vee \sigma_2$	2	$S_{\beta_1} = \{\sigma_1\}$ $S_{\beta_2} = \{\sigma_2\}$
$(\mathsf{A}\sigma)$	$\mathsf{A}(\sigma_1, \ldots, \sigma_n, \Pi)$	$n+1$	$S_{\beta_1} = \{\sigma_1\}$ $\vdots$ $S_{\beta_n} = \{\sigma_n\}$ $S_{\beta_{n+1}} = \{\mathsf{A}\Pi\}$
$(\mathsf{E}\Box\sigma)$	$\mathsf{E}(\Box(\sigma_1 \vee \Box\sigma_2), \Pi)$	2	$S_{\beta_1} = \{\sigma_1, \mathsf{E}(\circ\Box(\sigma_1 \vee \Box\sigma_2), \Pi)\}$ $S_{\beta_2} = \{\neg\sigma_1, \sigma_2, \mathsf{E}(\circ\Box\sigma_2, \Pi)\}$
$(\mathsf{E}\mathcal{U}\sigma)$	$\mathsf{E}(\sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3), \Pi)$	2	$S_{\beta_1} = \{\sigma_2, \mathsf{E}(\Diamond\sigma_3, \Pi\}$ $S_{\beta_2} = \{\sigma_1, \mathsf{E}(\circ(\sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3)), \Pi)\}$
$(\mathsf{E}\mathcal{U}\Box)$	$\mathsf{E}(\sigma_1 \mathcal{U} \Box\sigma_2, \Pi)$	2	$S_{\beta_1} = \{\mathsf{E}(\Box\sigma_2, \Pi)\}$ $S_{\beta_2} = \{\sigma_1, \mathsf{E}(\circ(\sigma_1 \mathcal{U} \Box\sigma_2), \Pi)\}$
$(\mathsf{E}\Box\mathcal{U})$	$\mathsf{E}(\Box(\sigma_1 \mathcal{U} \sigma_2), \Pi)$	2	$S_{\beta_1} = \{\sigma_2, \mathsf{E}(\circ\Box(\sigma_1 \mathcal{U} \sigma_2), \Pi)\}$ $S_{\beta_2} = \{\sigma_1, \mathsf{E}(\circ\Box(\sigma_1 \mathcal{U} \sigma_2), \Pi)\}$
$(\mathsf{A}\Box\sigma)$	$\mathsf{A}(\Box(\sigma_1 \vee \Box\sigma_2), \Pi)$	3	$S_{\beta_1} = \{\sigma_1, \mathsf{A}(\circ\Box(\sigma_1 \vee \Box\sigma_2), \Pi)\}$ $S_{\beta_2} = \{\neg\sigma_1, \sigma_2, \mathsf{A}(\circ\Box\sigma_2, \Pi)\}$ $S_{\beta_3} = \{\mathsf{A}\Pi\}$
$(\mathsf{A}\mathcal{U}\sigma)$	$\mathsf{A}(\sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3), \Pi)$	3	$S_{\beta_1} = \{\sigma_2, \mathsf{A}(\Diamond\sigma_3, \Pi)\}$ $S_{\beta_2} = \{\sigma_1, \mathsf{A}(\circ(\sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3)), \Pi)\}$ $S_{\beta_3} = \{\mathsf{A}\Pi\}$
$(\mathsf{A}\mathcal{U}\Box)$	$\mathsf{A}(\sigma_1 \mathcal{U} \Box\sigma_2, \Pi)$	2	$S_{\beta_1} = \{\mathsf{A}(\Box\sigma_2, \Pi)\}$ $S_{\beta_2} = \{\sigma_1, \sigma_2, \mathsf{A}(\circ(\sigma_1 \mathcal{U} \Box\sigma_2), \Pi)\}$

Fig. 4: BETA RULES. (Notation: σ, σ_i stand for state formulae, π_i stand for path formulae, and Π is a (possibly empty) set of path formulae.)

3.1 Preliminaries

Definition 10 (Tableau, Consistent Node, Closed branch). *A tableau for a set of* $\mathsf{ECTL}^\#$ *state formulae* Σ *is a labelled tree* T*, where nodes are* τ*-labeled with sets of state formulae, such that the following two conditions hold:*

(a) The root is labelled by the set Σ.
(b) Any other node m is labelled with sets of state formulae as the result of the application of one of the rules in Figures 3, 4, 5 and 7 to its parent node n. Given the applied rule is R, we term m an R-successor of n.

A node n of T is consistent, abbreviated as $n_\top$, if $\tau(n)$ is a syntactically consistent set of formulae (see Def. 4), else n is inconsistent, abbreviated as $n_\bot$. If a branch b of T, contains $n_\bot \in b$, then b is closed else b is open. ∎

$$(\mathsf{Q}\circ)\ \frac{\Sigma, \mathsf{A}\circ\Phi_1, \ldots, \mathsf{A}\circ\Phi_\ell, \mathsf{E}\circ\Psi_1, \ldots, \mathsf{E}\circ\Psi_k,}{\mathsf{A}\Phi_1, \ldots, \mathsf{A}\Phi_\ell, \mathsf{E}\Psi_1 \ \&\ \ldots\ \&\ \mathsf{A}\Phi_1, \ldots, \mathsf{A}\Phi_\ell, \mathsf{E}\Psi_k}$$

Fig. 5: NEXT-STATE RULE. (Notation: Σ is a (possibly empty) set of literals, and Φ_i, Ψ_i are non-empty sets of formulae.)

To make the presentation more transparent we give an informal overview of the tableau construction. Any tableau has a root-node that is exclusively labelled by a set of state formulae. To extend a node we apply one of α, β or β^+ rule. The first two types of rules are standard to the tableau, and are essentially based on the fixpoint characterisation of $\mathsf{Q}\Box$ and $\mathsf{Q}\mathcal{U}$ modalities, while β^+ rules are characteristic (and crucial!) for our construction. They tackle difficult cases of formulae in $\mathsf{ECTL}^{\#}$, and are related to our dedicated account of the eventualities. Namely, we treat an eventuality as occurring in some *context*, which, in turn, is a collection of all state formulae, called 'an outer context' or path formulae called 'an inner context'. As we will see, β^+ rules use the context to force eventualities to be fulfilled as soon as possible.

The $\alpha - \beta - \beta^+$ rules apply repeatedly until they produce an inconsistent node $n_\bot$, or a node with the labels that already occurred within the path under consideration. In the former case the expansion of the given branch terminates with $n_\bot$ as its leaf. In the latter case, a repetitive node in the branch suggests that the input formula is satisfied forever, and we select another eventuality (if any) see §4.1. Obviously, $n_\bot$ has an unsatisfiable $\tau(n)$ and is a 'deadlock' in the construction of a model. However, open branches do not necessarily give us a model. In particular, an open branch could be a prefix of a closed one. Later we introduce the notion of an expanded branch that enables the model construction. Once no more expansion rules are applicable to the given branch with the last node $n_\top$, we are ensured that $\tau(n) = \Sigma, \mathsf{A}\circ\Phi_1, \ldots, \mathsf{A}\circ\Phi_n, \mathsf{E}\circ\Psi_1, \ldots, \mathsf{E}\circ\Psi_m$, where Σ is a set of literals. This labelling $\tau(n)$ is similar to a 'state' in the standard temporal tableau. Then the 'next-state' rule applies to generate the successors of n with the labels that are arguments of all $\mathsf{Q}\circ$ modalities. The whole cycle of applying $\alpha - \beta - \beta^+$ and 'next-state' rules is repeated until the tableau construction terminates. The nature of our rules ensures that the terminated tableau represents a model for the tableau input if all the leaves in a collection of branches, called a bunch, are consistent and all eventualities occurring in looping branches are fulfilled, otherwise, the tableau input is unsatisfiable.

3.2 Alpha, Beta Rules and Next-State Rule

The α- and β-rules are the most elementary rules of our tableau system. An α-rule enlarges a branch with a node labelled by Σ, α, by a successor node labelled by Σ, S_α, where S_α is the set of formulae associated with α in Figure 3. An α-rule is represented as $\dfrac{\Sigma, \alpha}{\Sigma, S_\alpha}$ while β-rules as $\dfrac{\Sigma, \beta}{\Sigma, S_{\beta_1} \mid \cdots \mid \Sigma, S_{\beta_k}}$. β-rule splits a branch containing a node labelled by a set Σ, β (where β is one of the formulae of Figure 4), in k new nodes each labelled by the corresponding Σ, S_{β_i}, according to Figure 4. The next-state rule $(\mathsf{Q}\circ)$, Figure 5, also splits the branch into k branches each of them rooted by node n labelled by a set $\mathsf{A}\Phi_1, \ldots, \mathsf{A}\Phi_\ell, \mathsf{E}\Psi_i$, for $i \in \{1, \ldots, k\}$. This is the only rule of our system that splits branches in a *conjunctive way*. We use the symbol $\&$ to represent the generation of AND-successors of node n. When $\ell = k = 0$, the rule yields a unique new node labelled by the empty set. We assume that whenever $k = 0$ and $\ell > 0$, there exists a unique descendant labelled by $\mathsf{A}\Phi_1, \ldots, \mathsf{A}\Phi_l$.

Example 11. Let n be a node such that $\tau(n) = \{a, \neg b, \mathsf{A}\circ c, \mathsf{E}\circ p, \mathsf{E}\circ\neg p, \mathsf{A}\circ\Box((\mathsf{E}\circ p) \wedge (\mathsf{E}\circ\neg p))\}$. Then the next-state rule $(\mathsf{Q}\circ)$ applies to n generating the following AND-successors of n: $\{\mathsf{A}c, p, \mathsf{A}\Box((\mathsf{E}\circ p) \wedge (\mathsf{E}\circ\neg p))\}$ and $\{\mathsf{A}c, \neg p, \mathsf{A}\Box((\mathsf{E}\circ p) \wedge (\mathsf{E}\circ\neg p))\}$. Note that $\mathsf{A}c$ requires the application of the β-rule $(\mathsf{A}\sigma)$ to be reduced to c. ∎

3.3 The Uniform Tableau

In this subsection we explain how to construct a tableau where leaves are labelled by sets of formulae of a specific form – *Uniform* sets of state formulae.

Definition 12 (Elementary Set of $\mathsf{ECTL}^{\#}$ State Formulae). *A set of $\mathsf{ECTL}^{\#}$ state formulae is elementary if and only if it is exclusively formed by literals and formulae of the form $Q\circ\Pi$.* ∎

Proposition 13. *Any set of $\mathsf{ECTL}^{\#}$ state formulae has a tableau T such that all its leaves are labelled by elementary sets of state formulae.*

Proof. Repeatedly apply to every expandable node any applicable α-rule or β-rule. ∎

Example 14. Figure 6 depicts a tableau with elementary leaves for $\mathsf{A}(\circ q, \Box r), \mathsf{E}(\circ r, q\,\mathcal{U}\,\Box\neg p), \mathsf{E}\circ q$. Recall that $\mathsf{A}(\circ q, \Box r)$ is an abbreviation of $\mathsf{A}(\circ q, \Box(r \vee \Box\mathbf{F}))$.

Definition 15 (Basic Path/State Formula and Uniform Set of Formulae). *Every $\mathsf{ECTL}^{\#}$ path formula of the type $\circ\sigma, \sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3),\ \sigma_1\,\mathcal{U}\,(\Box\sigma_2),\ \Box(\sigma_1 \vee \Box\sigma_2),\ \Box(\sigma_1\,\mathcal{U}\,\sigma_2)$ is called basic. Every state formula $Q\Pi$ where Π is a set of basic-path formulae is also called basic. A set of state formulae Σ is uniform iff Σ is exclusively formed by literals and basic state formulae, and Σ contains at most one E-conjunctive formula.* ∎

Proposition 16. *Any set of $\mathsf{ECTL}^{\#}$ state formulae Σ has a tableau T such that labels of all its leaves are uniform sets of state formulae. Moreover, each open branch of T contains exactly one application of* $(\mathsf{Q}\circ)$.

Proof. Use Proposition 13 to construct a tableau with all its leaves labelled by elementary sets of formulae. Then apply the rule $(\mathsf{Q}\circ)$, to any relevant node and, finally, repeatedly apply (to every expandable node) the rules $(\mathsf{E}\sigma)$, $(\mathsf{A}\sigma)$, $(\wedge)$, and $(\vee)$. ∎

Definition 17 (Uniform Tableaux). *For any set Σ of $\mathsf{ECTL}^{\#}$ state formulae, the tableau for Σ provided by Proposition 16 is denoted* Uniform_Tableau*(Σ).* ∎

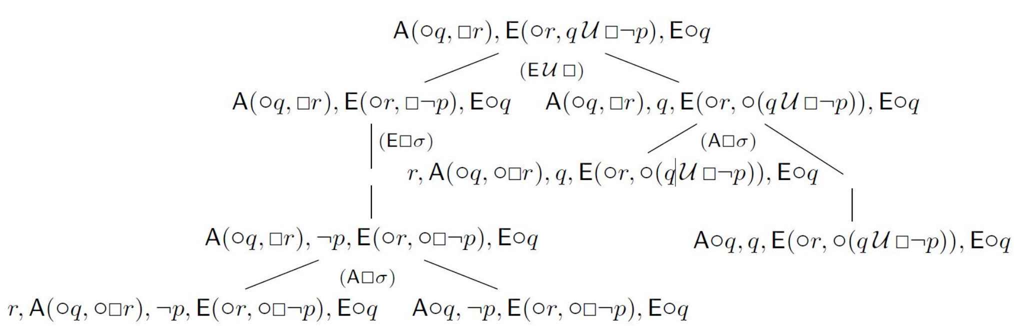

Fig. 6: A tableau whose leaves are elementary

Example 18. Constructing a uniform tableau for the set $\{\mathsf{A}(\circ q, \Box r), \mathsf{E}(\circ r, q\,\mathcal{U}\,\Box\neg p), \mathsf{E}\circ q)\}$, we first obtain the tableau in Figure 6. Then we apply the $(\mathsf{Q}\circ)$ rule enlarging each of the four branches and producing the following eight leaves, left to right (we refer to the node by its labels):

1. $\mathsf{A}(q, \Box r), \mathsf{E}(r, \Box\neg p)$ 2. $\mathsf{A}(q, \Box r), \mathsf{E}q$ 3. $\mathsf{A}q, \mathsf{E}(r, \Box\neg p)$ 4. $\mathsf{A}q, \mathsf{E}q$
5. $\mathsf{A}(q, \Box r), \mathsf{E}(r, q\,\mathcal{U}\,\Box\neg p)$ 6. $\mathsf{A}(q, \Box r), \mathsf{E}q$ 7. $\mathsf{A}q, \mathsf{E}(r, q\,\mathcal{U}\,\Box\neg p)$ 8. $\mathsf{A}q, \mathsf{E}q$

Then we apply the rules $(\mathsf{A}\sigma)$ and $(\mathsf{E}\sigma)$: the first branch is split into $q, r, \mathsf{E}\Box\neg p$ and $\mathsf{A}\Box r, r, \mathsf{E}\Box\neg p$; the second into q and $\mathsf{A}\Box r, q$; the third yields only a child $q, r, \mathsf{E}\Box\neg p$; the fourth and the eighth yield only q; the fifth is split into two nodes $q, r, \mathsf{E}(q\,\mathcal{U}\,\Box\neg p)$ and $\mathsf{A}\Box r, r, \mathsf{E}(q\,\mathcal{U}\,\Box\neg p)$; the sixth into q and $\mathsf{A}\Box r, q$; and the seventh yields the unique child $q, r, \mathsf{E}(q\,\mathcal{U}\,\Box\neg p)$. ■

β^+-Rule	Σ, β	k	$S^+_{\Sigma,\beta_i} (1 \le i \le k)$
$(\mathsf{A}\,\mathcal{U}\,\sigma)^+$	$\Sigma, \mathsf{A}(\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3), \Pi)$	3	$S^+_{\Sigma,\beta_1} = \{\sigma_2, \mathsf{A}(\Diamond\sigma_3, \Pi)\}$ $S^+_{\Sigma,\beta_2} = \{\sigma_1, \mathsf{A}(\circ((\sigma_1 \wedge (\neg\Sigma \vee \varphi_\Pi))\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)), \Pi)\}$ $S^+_{\Sigma,\beta_3} = \{\mathsf{A}\Pi\}$
$(\mathsf{A}\,\mathcal{U}\,\Box)^+$	$\Sigma, \mathsf{A}(\sigma_1\,\mathcal{U}\,\Box\sigma_2, \Pi)$	2	$S^+_{\Sigma,\beta_1} = \{\mathsf{A}(\Box\sigma_2, \Pi)\}$ $S^+_{\Sigma,\beta_2} = \{\sigma_1, \mathsf{A}(\circ((\sigma_1 \wedge (\neg\Sigma \vee \varphi_\Pi \vee \sigma_2))\,\mathcal{U}\,\Box\sigma_2), \Pi)\}$

β^+-Rule	Σ, β	k	$S^+_{\Sigma,\beta_i} (1 \le i \le k)$
$(\mathsf{E}\,\mathcal{U}\,\sigma\Box)^+$	$\Sigma, \mathsf{E}\Pi$	$2n$	S^+_{Σ,β_1} $\vdots$ $S^+_{\Sigma,\beta_i} \begin{cases} \{\sigma_2, \mathsf{E}(\Diamond\sigma_3, \Pi^{-i})\} & \text{if } \pi_i = \sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3) \\ \{\mathsf{E}(\Box\sigma_2, \Pi^{-i})\} & \text{if } \pi_i = \sigma_1\,\mathcal{U}\,\Box\sigma_2 \end{cases}$ $\vdots$ S^+_{Σ,β_n} $S^+_{\Sigma,\beta_{n+1}}$ $\vdots$ $S^+_{\Sigma,\beta_{2i}} \begin{cases} \{\sigma_1, \mathsf{E}(\circ((\sigma_1 \wedge \neg\Sigma)\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)), \Pi^{-i})\} & \text{if } \pi_i = \sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3) \\ \{\sigma_1, \mathsf{E}(\circ((\sigma_1 \wedge \neg\Sigma)\,\mathcal{U}\,\Box\sigma_2), \Pi^{-i})\} & \text{if } \pi_i = \sigma_1\,\mathcal{U}\,\Box\sigma_2 \end{cases}$ $\vdots$ $S^+_{\Sigma,\beta_{2n}}$

Fig. 7: BETA-PLUS RULES. (Notation: σ, σ_i stand for state formulae, Σ is a (possibly empty) set of state formulae, Π is a (possibly empty) set of basic-path formulae. Formula φ_Π is defined in Def. 19. We denote by $\Pi_{\mathcal{U}}$ the set of all formulae in Π that have the forms $\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$ and $\sigma_1\,\mathcal{U}\,\Box\sigma_2$. $\Pi_{\mathcal{U}}$ is enumerated as $\{\pi_1, \ldots, \pi_n\}$ for $n \ge 1$, and $\Pi^{-i} = \Pi \setminus \{\pi_i\}$.)

3.4 The Beta-plus Rules

In this subsection we extend our set of tableau rules with the new four rules named as β^+-rules (Figure 7). These rules, similarly to β-rules, also split a branch, but this time into a number of branches depending on the treated formula. The rules for A-disjunctive formulae apply to a label Σ, β, where β has the form $\mathsf{A}(\pi, \Pi)$, Π is a set of basic-path formulae, and π is either $\sigma\,\mathcal{U}\,(\sigma \wedge \Diamond\sigma)$ or $\sigma\,\mathcal{U}\,\Box\sigma$. The rule $(\mathsf{E}\,\mathcal{U}\,\sigma\Box)^+$ for E-conjunctive formulae applies to a set Σ, β, where β has the form $\mathsf{E}\Pi$ and Π is a set of basic-path formulae that contains at least one formula $\sigma\,\mathcal{U}\,(\sigma \wedge \Diamond\sigma)$ or $\sigma\,\mathcal{U}\,\Box\sigma$. The β^+-rules are the only rules in our system that make use of the so-called context for forcing the eventualities to be satisfied as soon as possible. The context is given by the sets Σ containing state formulae and Π containing path formulae. We name Σ the outer context and Π the inner context. The outer context is used by all the β^+-rules. The inner context is only needed to deal with formulae $\mathsf{A}\Pi$. The following formula, φ_Π, introduced in Def. 19 is used to manage the inner context Π in rules $(\mathsf{A}\,\mathcal{U}\,\sigma)^+$ and $(\mathsf{A}\,\mathcal{U}\,\Box)^+$.

Definition 19 (Formula φ_Π for β^+-rules on A-disjunctive formulae). *Let Π be a set of basic path formulae. We define the formula φ_Π to be the following disjunction of state formulae (φ_Π is $\mathbf{F}$, if the below disjunction is empty):*

$$\bigvee_{\Box(\sigma_1 \vee \Box\sigma_2)\in\Pi} (\sigma_1 \vee \sigma_2) \;\vee \bigvee_{\sigma_1\,\mathcal{U}\,\Box\sigma_2\in\Pi} \sigma_2 \;\vee \bigvee_{\Box(\sigma_1\,\mathcal{U}\,\sigma_2)\in\Pi} \mathsf{E}(\Diamond\sigma_2).$$

■

It is worth noting that each β^+-rule, when applied to some formula of the form $\mathsf{Q}(\sigma_1\,\mathcal{U}\,\varphi, \Pi)$ –where φ could be $\sigma_2 \wedge \Diamond\sigma_3$ or $\Box\sigma_2$– generates one or more successors that contain a formula of the form $\mathsf{Q}(\circ((\sigma_1 \wedge \sigma)\,\mathcal{U}\,\varphi), \Pi)$ where σ depends on both the inner and the outer context, and is defined depending on whether Q is E or A. We call $(\sigma_1 \wedge \sigma)\,\mathcal{U}\,\varphi$ the *next-step variant* of $\sigma\,\mathcal{U}\,\varphi$. Example 20 illustrates the main ideas behind the application of β^+-rules $(\mathsf{A}\,\mathcal{U}\,\sigma)^+$ and $(\mathsf{A}\,\mathcal{U}\,\sigma)$.

Example 20. The β^+-rules $(\mathsf{A}\,\mathcal{U}\,\sigma)^+$ applies to one selected formula with exactly one marked eventuality. Consequently, the $(\mathsf{A}\,\mathcal{U}\,\sigma)$-rule applies to all the eventualities (in the selected formula) except to the marked one.

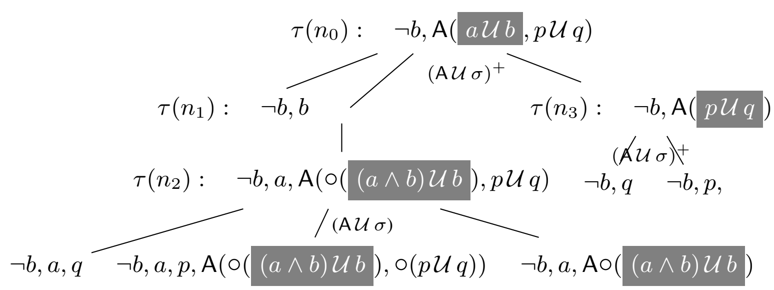

Fig. 8: Application of $(\mathsf{A}\,\mathcal{U}\,\sigma)^+$ and $(\mathsf{A}\,\mathcal{U}\,\sigma)$ rules to $\{\neg b, \mathsf{A}(a\,\mathcal{U}\,b, p\,\mathcal{U}\,q)\}$

In Figure 8 the marked eventualities are in gray boxes. Assume $a\,\mathcal{U}\,b$ is the marked eventuality. Then, the $(\mathsf{A}\,\mathcal{U}\,\sigma)^+$-rule can be applied to $a\,\mathcal{U}\,b)$ with outer context $\Sigma = \{\neg b\}$ and inner context, $\Pi = \{p\,\mathcal{U}\,q\}$. According to Definition 19, φ_Π is $\mathbf{F}$. Therefore, $S^+_{\Sigma,\beta_1} = \{b\}$, $S^+_{\Sigma,\beta_3} = \{\mathsf{A}(p\,\mathcal{U}\,q)\}$, and $S^+_{\Sigma,\beta_2} = \{a, \mathsf{A}(\circ((a \wedge b)\,\mathcal{U}\,b), p\,\mathcal{U}\,q)\}$ respectively generate nodes n_1, n_2 and n_3. In node n_2, the $(\mathsf{A}\,\mathcal{U}\,\sigma)$-rule is applied to $p\,\mathcal{U}\,q$. That produces three new nodes. Regarding node n_3, the marked eventuality disappears. Then, a new selection is made and $p\,\mathcal{U}\,q$ is marked. Consequently, $(\mathsf{A}\,\mathcal{U}\,\sigma)^+$-rule is applied with outer context $\Sigma = \{\neg b\}$. The inner context is the empty set. Note that all leaves in Figure 8 are elementary. Hence, the construction of the tableau, would continue applying the next-state rule $(\mathsf{Q}\circ)$. ■

3.5 Simplification Rules

A large set of simplification rules can be used to reduce the tableau construction. Here we only mention those that are essential for termination. First, to stop the growth of the subformula σ in the successive next-step variants $(\sigma_1 \wedge \sigma)\,\mathcal{U}\,\varphi$, we use trivial simplification rules such as $\varphi \wedge \varphi \longrightarrow \varphi$ and $\varphi \vee \varphi \longrightarrow \varphi$, as well as classical subsumptions rules. Second, to simplify the detection of equal labels (for looping in tableau branches) we use the following (subsumption-based) rules:

$(\sqsubset \mathsf{E}\Box\mathcal{U})$ $\mathsf{E}(\sigma_1\,\mathcal{U}\,\sigma_2, \Box(\sigma_1\,\mathcal{U}\,\sigma_2), \Pi) \longrightarrow \mathsf{E}(\Box(\sigma_1\,\mathcal{U}\,\sigma_2), \Pi)$.
$(\sqsubset \mathsf{A}\Box\mathcal{U})$ If $\Pi' \subseteq \Pi$ then $\mathsf{A}(\sigma_1\,\mathcal{U}\,\sigma_2, \Pi) \wedge \mathsf{A}(\Box(\sigma_1\,\mathcal{U}\,\sigma_2), \Pi') \longrightarrow \mathsf{A}(\Box(\sigma_1\,\mathcal{U}\,\sigma_2), \Pi')$.

Finally, to prevent the duplications of the original eventuality $\sigma_1\,\mathcal{U}\,\sigma_2$ and its successive new-step variants by rules $(\mathsf{Q}\Box\mathcal{U})$ and $(\mathsf{Q}\,\mathcal{U}\,\sigma)^+$, and to ensure termination, we use the following (subsumption-based) simplification rules:

$(\sqsubset \mathsf{A}\sigma\mathcal{U})$ $\sigma_2 \wedge \mathsf{A}(\sigma_1\,\mathcal{U}\,\sigma_2, \Pi) \longrightarrow \sigma_2$.
$(\sqsubset \mathsf{E}\mathcal{U}\sigma)$ $\mathsf{E}((\sigma_1 \wedge \sigma)\,\mathcal{U}\,\varphi, \sigma_1\,\mathcal{U}\,\varphi, \Pi) \longrightarrow \mathsf{E}((\sigma_1 \wedge \sigma)\,\mathcal{U}\,\varphi, \Pi)$
$(\sqsubset \mathsf{A}\mathcal{U}\sigma)$ If $\Pi' \subseteq \Pi$ then $\mathsf{A}((\sigma_1 \wedge \sigma)\,\mathcal{U}\,\varphi, \Pi') \wedge \mathsf{A}(\sigma_1\,\mathcal{U}\,\varphi, \Pi) \longrightarrow \mathsf{A}((\sigma_1 \wedge \sigma)\,\mathcal{U}\,\varphi, \Pi')$.

3.6 The role of φ_{Π} in the Beta-plus Rules

Let us consider a set of formulae $\Phi = \Sigma, \mathsf{A}(\sigma \,\mathcal{U}\, \varphi, \Pi)$. A model, $\mathcal{K}$, of Φ could satisfy $\mathsf{A}(\sigma \,\mathcal{U}\, \chi, \Pi)$ because each fullpath of $\mathcal{K}$ satisfies either $\sigma \,\mathcal{U}\, \chi$ or some formula $\pi \in \Pi$. The next-step variant of $\sigma \,\mathcal{U}\, \chi$ is $\circ(\sigma \wedge (\neg\Sigma \vee \varphi_{\Pi})) \,\mathcal{U}\, \chi)$, which makes $\neg\Sigma$ or φ_{Π} satisfiable before χ is satisfied. The former produces open branches where χ is fulfilled as soon as possible, whereas the latter produces open branches that satisfy some of the $\pi \in \Pi$. Therefore, φ_{Π} allows to generate a model from a branch in which $\sigma \,\mathcal{U}\, \varphi$ is not fulfilled, and some $\pi \in \Pi$ is satisfied. Example 21 illustrates these ideas from the constructive view, i.e when we construct a tableau for a formula $\mathsf{A}(\pi_1, \ldots, \pi_n)$.

$$
\begin{array}{rl}
\tau(n_1): & \mathsf{A}(a \,\mathcal{U}\, b, \Box c, r \,\mathcal{U}\, \Box s, \Box(p \,\mathcal{U}\, q)) \\
 & \mid \ (\mathsf{A}\mathcal{U}\,\sigma)^+ \\
\tau(n_2): & a, \mathsf{A}(\circ(\overbrace{(a \wedge (c \vee s \vee \mathsf{E}\Diamond q)) \,\mathcal{U}\, b}^{\alpha_1}), \Box c, r \,\mathcal{U}\, \Box s, \Box(p \,\mathcal{U}\, q)) \\
 & \mid \ (\mathsf{A}\Box\sigma) + (\mathsf{A}\mathcal{U}\,\Box) + \mathsf{A}\Box\mathcal{U}\,) + (\mathsf{A}\mathcal{U}\,\sigma) + (\mathsf{Q}\circ) + (\sqsubset \mathsf{A}\Box\mathcal{U}\,) \\
\tau(n_3): & \mathsf{A}(\alpha_1, \Box c, (r \,\mathcal{U}\, \Box s), \Box(p \,\mathcal{U}\, q)) \\
 & \mid \ (\mathsf{A}\mathcal{U}\,\sigma)^+ \\
\tau(n_4): & a, (a \wedge (c \vee s \vee \mathsf{E}\Diamond q)), \mathsf{A}(\circ(\alpha_1), \Box c, r \,\mathcal{U}\, \Box s, \Box(p \,\mathcal{U}\, q)) \\
 & \mid \ (\mathsf{A}\Box\sigma) + (\mathsf{A}\mathcal{U}\,\Box) + \mathsf{A}\Box\mathcal{U}\,) + (\mathsf{A}\mathcal{U}\,\sigma) + (\wedge) + (\vee) \\
\tau(n_5): & a, q, c, r, p, \mathsf{A}(\circ(\alpha_1), \circ\Box c, \circ(r \,\mathcal{U}\, \Box s), \circ(p \,\mathcal{U}\, q)), \mathsf{A}(\circ(\alpha_1), \circ\Box c, \circ(r \,\mathcal{U}\, \Box s), \circ\Box(p \,\mathcal{U}\, q)) \\
 & \mid \ (\mathsf{Q}\circ) + (\sqsubset \mathsf{A}\Box\mathcal{U}\,) \\
\tau(n_6): & \mathsf{A}(\alpha_1, \Box c, (r \,\mathcal{U}\, \Box s), \Box(p \,\mathcal{U}\, q))
\end{array}
$$

Fig. 9: A branch of a tableau for $\mathsf{A}(a \,\mathcal{U}\, b, \Box c, r \,\mathcal{U}\, \Box s, \Box(p \,\mathcal{U}\, q)$.

Example 21. Let $\Pi = \{\Box c, r \,\mathcal{U}\, \Box s, \Box(p \,\mathcal{U}\, q)\}$, and consider an application of the rule $(\mathsf{A}\mathcal{U}\,\sigma)^+$ to the formula $\mathsf{A}(a \,\mathcal{U}\, b, \Pi)$, where $a, b, c, p, q, r, s \in \mathsf{Prop}$ (see Figure 9). The outer context, namely Σ, is empty and the inner context is Π. Then $\neg\Sigma$ is $\mathbf{F}$ and $\varphi_{\Pi} = c \vee s \vee \mathsf{E}\Diamond q$. Hence, the second child, namely S^+_{Σ,β_2}, raised by the application of $(\mathsf{A}\mathcal{U}\,\sigma)^+$ is labelled by $\{a, \mathsf{A}(\circ\alpha_1, \Pi)\}$ where $\alpha_1 = (a \wedge \varphi_{\Pi}) \,\mathcal{U}\, b = (a \wedge (c \vee s \vee \mathsf{E}\Diamond q) \,\mathcal{U}\, b$. Then, Uniform_Tableau applies the (corresponding) rules to $\Box c, r \,\mathcal{U}\, \Box s, \Box(p \,\mathcal{U}\, q)$, and, finally, $(\mathsf{Q}\circ)$ and the simplification rule $(\sqsubset \mathsf{A}\Box\mathcal{U}\,)$, obtaining the node n_3. Now, one of the leaves raised by Uniform_Tableau is the fifth node; by applying here $(\mathsf{Q}\circ)$ and $(\sqsubset \mathsf{A}\Box\mathcal{U}\,)$ we obtain the last node n_6 such that $\tau(n_6) = \tau(n_3)$. This branch represents a model of the initial A-disjunctive formula where both disjuncts $\Box(p \,\mathcal{U}\, q)$ and $\Box c$ are satisfied, though the other two disjuncts are not. Indeed, it represents the model $\{a, c, r, p\}, (\{a, c, r, p, q\})^{\omega}$. ∎

4 Systematic Tableau Construction

In this section we define an algorithm, $\mathcal{A}^{sys}$, that constructs a *systematic tableau* and illustrate its performance with some examples. Recall that due to the rule $(\mathsf{Q}\circ)$, any open tableau should have a collection of open branches including all the $(\mathsf{Q}\circ)$-successors of any node labelled by an elementary set of formulae. These collections of branches are called *bunches*. Any open bunch of the systematic tableau, constructed by the algorithm $\mathcal{A}^{sys}$ introduced in this section, enables the construction of a model for the initial set of formulae.

Algorithm 1 Systematic Tableau Construction

procedure SYSTEMATIC_TABLEAU(Σ_0) $\quad\triangleright$ where Σ_0: set of state formulae
  if Σ_0 is not uniform **then** T := Uniform_Tableau(Σ_0)
  while T has at least one expandable node **do**
    $\triangleright$ Invariant: Any expandable node of T is labelled by an uniform set
    Choose any node ℓ in T such that $\tau(\ell)$ is expandable
    Let $\Sigma = \tau(\ell)$ $\quad\triangleright$ Σ is uniform
    if there are not selectable formulae in ℓ **then** T := $T[\ell \leftarrow$Uniform_Tableau(Σ)]
    else
      Eventuality_Selection(Σ)
      Apply_β^+-rule(Σ)
      Let k be the number of new leaves, $\ell_1, \dots, \ell_k$ the new leaves and $\Sigma_1, \dots, \Sigma_k$ their labels
      for i = 1 .. k **do**
        if ℓ_i is expandable and Σ_i is not uniform **then**
          T := $T[\ell_i \leftarrow$Uniform_Tableau(Σ_i)]
  return T

4.1 The Algorithm

The algorithm $\mathcal{A}^{sys}$ constructs an *expanded* tableau (see Definition 41) for the given input. $\mathcal{A}^{sys}$ applied to the input Σ_0, denoted as $\mathcal{A}^{sys}(\Sigma_0)$, returns a systematic tableau $\mathcal{A}^{sys}_{\Sigma_0}$. Intuitively, 'expanded' means 'complete' in the sense that any possible rule has been already applied at every node. Though the best way to implement this algorithm is a depth-first construction, for clarity, we formulate it as a breadth-first construction of a collection of subtrees. The procedure Uniform_Tableau, in the above Algorithm 1, was introduced in Definition 17 along with the notion of a uniform set of state formulae. The notation $T_1[\ell \leftarrow T_2]$ stands for the tableau T_1 where the expandable ℓ is substituted by the tableau T_2. In particular, $T[\ell \leftarrow$Uniform_Tableau(Σ)] is the tableau T where the expandable ℓ is substituted by the Uniform_Tableau(Σ). Next, we define the other two auxiliary procedures: Eventuality_Selection and Apply_β^+-rule, as well as the related concepts of *selectable formula*, *non-expandable* node and *eventuality-covered* branch. From now on any basic path formula that is either $\sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3)$ or $\sigma_1 \mathcal{U} \Box\sigma_2$ or $\Box(\sigma_1 \mathcal{U} \sigma_2)$ is called an *eventuality*. It is worth noting that $\circ\pi$ is not called an eventuality in this setting.

Definition 22 (Selectable Formula). *A formula is selectable if it is a $\mathsf{Q}\Pi$ and Π contains at least one eventuality.* ∎

Procedure Eventuality_Selection chooses formula $\mathsf{Q}\Pi$ and if $\mathsf{Q} = \mathsf{A}$ then the procedure also marks one eventuality, according to the priorities of selection and marking in Definition 24. We denote by $\pi_{\mathcal{U}}$ the marked eventuality in the selected formula $\mathsf{A}\Pi$.

Procedure Apply_β^+-rule(Σ) applies the corresponding rules depending on the selected formula $\mathsf{Q}\Pi$ and on the marked eventuality in the case $\mathsf{Q} = \mathsf{A}$:

- If $\mathsf{Q} = \mathsf{A}$ and $\sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3) \in \Pi$ is the marked eventuality, then apply $(\mathsf{A}\mathcal{U}\sigma)^+$
- If $\mathsf{Q} = \mathsf{A}$ and $\sigma_1 \mathcal{U} \Box\sigma_2 \in \Pi$ is the marked eventuality, then apply $(\mathsf{A}\mathcal{U}\Box)^+$
- If $\mathsf{Q} = \mathsf{A}$ and $\Box(\sigma_1 \mathcal{U} \sigma_2) \in \Pi$ is the marked eventuality, then first apply the rule $(\mathsf{A}\Box\mathcal{U})$ and then the rule $(\mathsf{A}\mathcal{U}\sigma)^+$ with the marked eventuality $\sigma_1 \mathcal{U} \sigma_2$.
- If $\mathsf{Q} = \mathsf{E}$ and Π contains at least one $\sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3)$ or one $\sigma_1 \mathcal{U} \Box\sigma_2$ then apply $(\mathsf{E}\mathcal{U}\sigma\Box)^+$

- If $\mathsf{Q} = \mathsf{E}$ and Π contains at least one $\Box(\sigma_1 \mathcal{U}\, \sigma_2)$ (but none $\sigma_1 \mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$ and none $\sigma_1 \mathcal{U}\, \Box\sigma_2$), then first apply the rule $(\mathsf{E}\Box\mathcal{U}\,)$ to every $\Box(\sigma_1 \mathcal{U}\, \sigma_2)$ and then the rule $(\mathsf{E}\mathcal{U}\,\sigma\Box)^+$.

Each application of a β^+-rule on the selected $\mathsf{A}\Pi$ introduces a next-step variant of the marked eventuality and each application of a β^+-rule on the selected $\mathsf{E}\Pi$ introduces a next-step variant for each $\sigma_1 \mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$ and each $\sigma_1 \mathcal{U}\, \Box\sigma_2$.

The call $\mathsf{Eventuality\_Selection}(\Sigma)$ keeps the selection of the formula $\mathsf{E}\Pi$ such that Π contains at least one $\sigma \mathcal{U}\,(\sigma \wedge \Diamond\sigma)$ or $\sigma \mathcal{U}\, \Box\sigma$, or keeps the selection of the formula $\mathsf{A}\Pi \in \Sigma$ which contains the next-step variant of the marked eventuality, or selects a new formula $\mathsf{A}\Pi \in \Sigma$ that contains an eventuality. The latter can happen for three reasons. When formulae $\mathsf{E}\Pi$ do not contain any $\sigma \mathcal{U}\,(\sigma \wedge \Diamond\sigma)$ nor $\sigma \mathcal{U}\, \Box\sigma$, or there is no marked eventuality in formulae $\mathsf{A}\Pi$, or when there is one, the node $\ell, (\Sigma = \tau(\ell))$ is a loop-node (see Definition 25), and the branch from the root-node to ℓ is not *eventuality-covered* (see Definition 26). In this case, a new selection should be made because there are eventualities that have never been marked but they should be. This way we introduce the term *eventuality-covered*. When a branch is eventuality-covered, its leaf is a loop-node and we are sure that, along the loop, at least some eventuality in each A-disjunctive formula and all eventualities in each E-conjunctive formula have been fulfilled. Consequently, the branch is an expanded open branch (see Definition 41) and represents a path in a possible model. It is worth mentioning that the only requirement for a branch to be eventuality-covered is to mark all necessary eventualities. The fact that formulae $\mathsf{E}\Pi$ are kept selected whereas they contain some eventuality and formulae $\mathsf{A}\Pi$ are kept selected where the next-step variant of the marked eventuality is kept marked ensures that every eventuality in $\mathsf{E}\Pi$ and at least one in each $\mathsf{A}\Pi$ is fulfilled.

When making the selection, priorities are used as stated in Definition 24. The idea behind priorities is that a tableau branch represents a path in a cyclic Kripke structure that is a possible model for the input formula. Therefore, it consists of a (possibly empty) initial sequence of states followed by a looping-sequence. Selectable formulae are classified into two sets - those of the highest priority and those of the lowest priority. The non-looping initial sequence is the first part of the model, hence we firstly select formulae $\mathsf{A}\Pi$ where Π is exclusively formed by formulae of the form $\sigma_1 \mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$ and formulae $\mathsf{E}\Pi$ where Π contains at least one eventuality of the form $\sigma_1 \mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$ or $\sigma_1 \mathcal{U}\, \Box\sigma_2$. These are the highest priority formulae, which cannot produce a loop. When one of the former formulae $\mathsf{A}\Pi$ is selected, one of the $\sigma_1 \mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$ is marked, namely π. By means of the rule $(\mathsf{A}\mathcal{U}\,\sigma)^+$, in a finite number of steps, either the branch close or π is either fulfilled (note that in the first branch $\mathsf{A}(\Diamond\sigma_3, \Pi)$ is also of the highest priority) or deleted (the third branch of $(\mathsf{A}\mathcal{U}\,\sigma)^+$). In any case the original formulae $\mathsf{A}\Pi$ disappears. When one of the latter formulae $\mathsf{E}\Pi$ is selected, the successive applications of the rule $(\mathsf{E}\mathcal{U}\,\sigma\Box)^+$ ensure (excluding the case when the branch closes) the fulfillment of all $\sigma_1 \mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$ or $\sigma_1 \mathcal{U}\, \Box\sigma_2$ in a finite number of states. Note that $\mathsf{E}(\Diamond\sigma_3, \Pi)$ is also of the highest priority. Once such formulae are fulfilled, all formulae $\sigma_1 \mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$ have disappeared from the E-conjunctive formula, whereas $\Box\sigma_2$ remains in the conjunction for all $\sigma_1 \mathcal{U}\, \Box\sigma_2 \in \Pi$. Hence, the residual $\mathsf{E}\Pi'$ is of the lowest-priority. On the contrary, the lowest priority formulae could produce a loop. They are formulae $\mathsf{A}\Pi$ where Π contains at least one $\sigma_1 \mathcal{U}\, \Box\sigma_2$ or $\Box(\sigma_1 \mathcal{U}\, \sigma_2)$ and formulae $\mathsf{E}\Pi$ where Π contains at least one $\Box(\sigma_1 \mathcal{U}\, \sigma_2)$ (but are not of the highest priority). They could produces a loop in a finite number of steps, since the subformulae starting by $\Box$ remains forever in the E-disjunctive formulae, whereas in the A-disjunctive formulae they can either remain or disappear. In the latter case, the residual A-disjunctive formulae could become non-selectable. It is easy to see that non-selectable formulae necessarily produce a loop.

Example 23. Consider $\Sigma_0 = \{\mathsf{A}(a\mathcal{U}\, b, b\mathcal{U}\, c), \mathsf{E}(p\mathcal{U}\, q, \Box(r\mathcal{U}\, s)), \mathsf{A}\Box(c\mathcal{U}\, d), \neg b, \mathsf{A}\Box e\}$. The first two formulae have the highest-priority, the third has the lowest priority, and the the last two are non-selectable. Suppose that we select $\mathsf{A}(a\mathcal{U}\, b, b\mathcal{U}\, c)$ and mark $a\mathcal{U}\, b$, since $\neg b \in \Sigma_0$, the left-most open branch of rule $(\mathsf{A}\mathcal{U}\,\sigma)^+$ contains $a, \mathsf{A}(\circ((a \wedge \neg\Sigma_0')\mathcal{U}\, b), b\mathcal{U}\, c)$ where $\Sigma_0' = \Sigma_0 \setminus \{\mathsf{A}(a\mathcal{U}\, b, b\mathcal{U}\, c)\}$. After applying the corresponding α and β rules to the remaining formulae, the first stage s_0 (the first state of the model) contains the atoms $\{a, q, s, d, e\}$. Then, by the next-step rule $(\mathsf{Q}\circ)$, the first node of the second stage s_1 is $\Sigma_1 = \{\mathsf{A}((a \wedge \neg\Sigma_0')\mathcal{U}\, b, b\mathcal{U}\, c), \mathsf{E}\Box(r\mathcal{U}\, s), \mathsf{A}\Box(c\mathcal{U}\, d), \mathsf{A}\Box e\}$ where the first formula is kept selected and the first eventuality is kept marked. Note that the second formula has now the lowest priority. Then we apply $(\mathsf{A}\mathcal{U}\,\sigma)^+$ to the first formulae and the corresponding α and β rules to the remaining formulae in Σ_1, generating the set of atoms in the left-most branch are $\{b, s, d, e\}$. Then, by the next-step rule $(\mathsf{Q}\circ)$, the first node of the third stage s_2 is $\Sigma_2 = \{\mathsf{E}\Box(r\mathcal{U}\, s), \mathsf{A}\Box(c\mathcal{U}\, d), \mathsf{A}\Box e\}$ where the first two formulas are of the lowest priority and the third is non-selectable. By selecting the first formula, the atoms in the stages s_2 (of the left-most branch) are $\{s, d, e\}$ and the new uniform set at the first node of the following stage is $\Sigma_3 = \Sigma_2$. However, $\mathsf{A}\Box(c\,\mathcal{U}\,d)$ has not been selected inside the loop, hence we produce one stage more, s_3, with atoms $\{s, d, e\}$. Then, by ($\mathsf{Q}\circ$), we obtain $\Sigma_4 = \Sigma_2$ and the branch is eventuality-covered. Therefore, we have a model for Σ_0 is $s_0, s_1\langle s_2, s_3\rangle^\omega$. ■

Definition 24 (Priorities for Eventuality Selection). *The selectable formulae of the highest priority for* Eventuality_Selection *are the formulae of the following two forms:*

- $\mathsf{A}\Pi$ *where* Π *is exclusively formed by formulae of the form* $\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$.
- $\mathsf{E}\Pi$ *where* Π *contains at least one eventuality of the form* $\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$ *or* $\sigma_1\,\mathcal{U}\,\Box\sigma_2$.

Consequently, the (selectable) formulae of the lowest priority are the formulae of the following two forms:

- $\mathsf{A}\Pi$ *where* Π *contains at least one* $\sigma_1\,\mathcal{U}\,\Box\sigma_2$ *or* $\Box(\sigma_1\,\mathcal{U}\,\sigma_2)$.
- $\mathsf{E}\Pi$ *where* Π *does not contain any* $\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$ *nor* $\sigma_1\,\mathcal{U}\,\Box\sigma_2$*, and* Π *contains at least one* $\Box(\sigma_1\,\mathcal{U}\,\sigma_2)$. ■

Once all the highest priority formulae have been selected in a branch, the only selectable formulae are the lowest priority formulae. At this point, the objective is to get a loop-node that makes the branch eventuality-covered.

Definition 25 (Loop-node). *Let* b *be a tableau branch and* $n_i \in b$ $(0 \leq i)$*. Then* n_i *is a loop-node if there exists* $n_j \in b$ $(0 \leq j < i)$ *and* $\tau(n_i) = \tau(n_j)$*. We say that* n_j *is a companion node of* n_i*.* ■

Definition 26 (Eventuality-covered Branch). *A tableau branch* $b = n_0, n_1, ..., n_i$ *is eventuality-covered if* n_i *is a loop-node, with a companion node* n_j $(0 \leq j < i)$*, both labelled by a uniform set* Σ *of non-selectable and the lowest priority formulae such that every selectable formula* $\mathsf{Q}\Pi \in \tau(n_i)$ *is selected in some node* n_k $(j \leq k < i)$ *and for every selected* $\mathsf{A}\Pi$ *exactly one eventuality in* Π *is marked in some node* n_k *such that* $j \leq k < i$*.* ■

The procedure Eventuality_Selection performs in some fair way that ensures that any open branch will ever be *eventuality-covered*.

Definition 27 (Non-expandable Node). *A node* n *is non-expandable, if* $\tau(n) = \Sigma_\perp$ *or* n *is a loop-node of branch* b *which is eventuality-covered. Otherwise,* n *is expandable.* ■

Consequently, an expandable node is either a node that is not a loop-node or a loop-node whose branch is not eventuality-covered. It is worth noting that a formula of the lowest priority could be selected more than once in a branch because the loop-node could change along the branch. In the following Example 28, we illustrate this issue.

Example 28. Figure 10 shows a branch in the systematic tableau for $\Sigma_0 = \{\mathsf{A}(p\,\mathcal{U}\,\neg p, \Box p), \mathsf{A}\Box(a\,\mathcal{U}\,\mathsf{E}\Box c)\}$ where, for readability, the marked eventualities are in gray boxes. The call Eventuality_Selection(Σ_0) selects the formula $\mathsf{A}(p\,\mathcal{U}\,\neg p, \Box p)$ in n_1. Generating n_2, when we apply the $(\mathsf{A}\,\mathcal{U}\,\sigma)^+$ rule to $\mathsf{A}(p\,\mathcal{U}\,\neg p, \Box p)$, the outer context is $\mathsf{A}\Box(a\,\mathcal{U}\,\mathsf{E}\Box c)$ and the inner context is p. Hence, in the S^+_{Σ,β_2} branch, the next-step variant of $p\,\mathcal{U}\,\neg p$ is $p \wedge (\neg(\mathsf{A}\Box(a\,\mathcal{U}\,\mathsf{E}\Box c) \vee p))\,\mathcal{U}\,\neg p$. By classical subsumption (included in our simplification rules), $p \wedge (\neg(\mathsf{A}\Box(a\,\mathcal{U}\,\mathsf{E}\Box c) \vee p) \;\longrightarrow\; p$, hence the next-step variant is $p\,\mathcal{U}\,\neg p$, and the formula $\mathsf{A}(\circ(p\,\mathcal{U}\,\neg p), \Box p)$ is added to the current stage. Then, applying $(\mathsf{A}\Box\sigma)$ to $\mathsf{A}(\circ(p\,\mathcal{U}\,\neg p), \Box p)$, $(\mathsf{A}\Box\,\mathcal{U}\,)$ to $(\mathsf{A}\Box(a\,\mathcal{U}\,\mathsf{E}\Box c)$, and $(\mathsf{A}\,\mathcal{U}\,\sigma)$ to $\mathsf{A}(a\,\mathcal{U}\,\mathsf{E}\Box c)$, the node n_2 is obtained. After the application of ($\mathsf{Q}\circ$) and ($\sqsubset$ $\mathsf{A}\Box\,\mathcal{U}$), n_3 is obtained. Node n_3 is a loop-node whose companion node is n_1 $(\tau(n_3) = \tau(n_1))$. However, the branch is not eventually-covered since the eventuality $a\,\mathcal{U}\,\mathsf{E}\Box c$ is not selected inside the loop. Therefore, we obtain n_4 by the call Eventuality_ Selection($(\tau(n_3))$ which selects $\mathsf{A}(a\,\mathcal{U}\,\mathsf{E}\Box c)$. After applying the $(\mathsf{A}\,\mathcal{U}\,\sigma)^+$ rule to $\mathsf{A}(a\,\mathcal{U}\,\mathsf{E}\Box c)$ (once the $(\mathsf{A}\Box\,\mathcal{U}\,)$ rule is applied), the branch S^+_{Σ,β_1} expands to n_5. After that, Uniform_Tableau gets one expandable node labelled by the uniform set $\{\mathsf{E}\Box c, \mathsf{A}(p\,\mathcal{U}\,\neg p, \Box p), \mathsf{A}\Box(a\,\mathcal{U}\,\mathsf{E}\Box c)\}$ as represented in n_6. The call Eventuality_Selection($(\tau(n_6))$ selects again the formula $\mathsf{A}(p\,\mathcal{U}\,\neg p, \Box p)$. Now, the outer context is $\{\mathsf{E}\Box c, \mathsf{A}\Box(a\,\mathcal{U}\,\mathsf{E}\Box c)\}$ and the inner context is p. Hence, by subsumption, $p \wedge (\neg(\mathsf{E}\Box c) \vee \neg(\mathsf{A}\Box(a\,\mathcal{U}\,\mathsf{E}\Box c)) \vee p) \;\longrightarrow\; p$. Hence, the S^+_{Σ,β_2} branch contains again the next-step variant $p\,\mathcal{U}\,\neg p$ in n_7. Then, expanding the Uniform_Tableau we obtain n_8 which

$$
\begin{array}{rl}
\tau(n_1): & \mathsf{A}(\boxed{p\,\mathcal{U}\,\neg p}, \Box p), \mathsf{A}\Box(a\,\mathcal{U}\,\mathsf{E}\Box c) \\
 & \quad | \; (\mathsf{A}\mathcal{U}\sigma)^{+} + (\mathsf{A}\Box\sigma) + (\mathsf{A}\Box\mathcal{U}) + (\mathsf{A}\mathcal{U}\sigma) \\
\tau(n_2): & p, a, \mathsf{A}(\bigcirc(\boxed{p\,\mathcal{U}\,\neg p}), \bigcirc\Box p), \mathsf{A}\bigcirc(a\,\mathcal{U}\,\mathsf{E}\Box c), \mathsf{A}\bigcirc\Box(a\,\mathcal{U}\,\mathsf{E}\Box c) \\
 & \quad | \; (\mathsf{Q}\bigcirc) + (\sqsubset \mathsf{A}\Box\mathcal{U}) \\
\tau(n_3): & \mathsf{A}(\boxed{p\,\mathcal{U}\,\neg p}, \Box p), \mathsf{A}\Box(a\,\mathcal{U}\,\mathsf{E}\Box c) \\
 & \quad | \; \text{(not eventuality covered)} \\
\tau(n_4) & \mathsf{A}(p\,\mathcal{U}\,\neg p, \Box p), \mathsf{A}\Box(\boxed{a\,\mathcal{U}\,\mathsf{E}\Box c}) \\
 & \quad | \; (\mathsf{A}\Box\mathcal{U}) + (\mathsf{A}\mathcal{U}\sigma)^{+} \\
\tau(n_5): & \mathsf{E}\Box c, \mathsf{A}(p\,\mathcal{U}\,\neg p, \Box p), \mathsf{A}\bigcirc\Box(a\,\mathcal{U}\,\mathsf{E}\Box c) \\
 & \quad | \; (\mathsf{A}\mathcal{U}\sigma) + (\mathsf{A}\Box\sigma) + (\mathsf{E}\Box\sigma) + (\mathsf{Q}\bigcirc) \\
\tau(n_6): & \mathsf{E}\Box c, \mathsf{A}(\boxed{p\,\mathcal{U}\,\neg p}, \Box p), \mathsf{A}\Box(a\,\mathcal{U}\,\mathsf{E}\Box c) \\
 & \quad | \; (\mathsf{A}\mathcal{U}\sigma)^{+} \\
\tau(n_7): & p, \mathsf{E}\Box c, \mathsf{A}(\bigcirc(\boxed{p\,\mathcal{U}\,\neg p}), \Box p), \mathsf{A}\Box(a\,\mathcal{U}\,\mathsf{E}\Box c) \\
 & \quad | \; (\mathsf{A}\Box\sigma) + (\mathsf{A}\Box\mathcal{U}) + (\mathsf{A}\mathcal{U}\sigma) + (\mathsf{E}\Box\sigma) \\
\tau(n_8): & p, c, \mathsf{E}\bigcirc\Box c, \mathsf{A}(\bigcirc(\boxed{p\,\mathcal{U}\,\neg p}), \bigcirc\Box p), \mathsf{A}\bigcirc\Box(a\,\mathcal{U}\,\mathsf{E}\Box c) \\
 & \quad | \; (\mathsf{Q}\bigcirc) \\
\tau(n_9): & \mathsf{E}\Box c, \mathsf{A}(\boxed{p\,\mathcal{U}\,\neg p}, \Box p), \mathsf{A}\Box(a\,\mathcal{U}\,\mathsf{E}\Box c) \\
 & \quad | \; \text{(not eventuality covered)} \\
\tau(n_{10}): & \mathsf{E}\Box c, \mathsf{A}(p\,\mathcal{U}\,\neg p, \Box p), \mathsf{A}\Box(\boxed{a\,\mathcal{U}\,\mathsf{E}\Box c}) \\
 & \quad | \; (\mathsf{A}\Box\mathcal{U}) + (\mathsf{A}\mathcal{U}\sigma)^{+} + (\mathsf{A}\mathcal{U}\sigma) + (\mathsf{A}\Box\sigma) + (\mathsf{E}\Box\sigma) \\
\tau(n_{11}): & p, c, \mathsf{E}\bigcirc\Box c, \mathsf{A}(\bigcirc(p\,\mathcal{U}\,\neg p), \bigcirc\Box p), \mathsf{A}\bigcirc\Box(a\,\mathcal{U}\,\mathsf{E}\Box c) \\
 & \quad | \; (\mathsf{Q}\bigcirc) \\
\tau(n_{12}): & \mathsf{E}\Box c, \mathsf{A}(p\,\mathcal{U}\,\neg p, \Box p), \mathsf{A}\Box(a\,\mathcal{U}\,\mathsf{E}\Box c)
\end{array}
$$

Fig. 10: A branch in the systematic tableau for $\mathsf{A}(p\,\mathcal{U}\,\neg p, \Box p), \mathsf{A}\Box(a\,\mathcal{U}\,\mathsf{E}\Box c)$

is an expandable loop-node because $\tau(n_8) = \tau(n_6)$. However, the branch is not yet eventually-covered since $a\,\mathcal{U}\,\mathsf{E}\square c$ has not been marked inside the loop. Then, the selected formula in n_9 is $\mathsf{A}(a\,\mathcal{U}\,\mathsf{E}\square c)$. Finally, Uniform_Tableau obtains a non-expandable loop-node, thus the given branch is eventually-covered - depicted in Figure 10 it represents $\{p,a\}(\{p,c\})^w$, which is a model of Σ_0. However, considering all the nodes in the branch, one would realize that the model represented is $\{p,a\}\{p,c\}(\{p,a,c\}\{p,c\})^w$. ∎

Definition 29 (Bunch in a Tableau, Closed Bunch and Tableau). *A bunch b is a collection of branches that is maximal with respect to $(\mathsf{Q}\circ)$-successor, i.e. every $(\mathsf{Q}\circ)$-successor of any node in b is also in b. A bunch is closed* iff *at least one of its branches is closed, otherwise it is open. A tableau is closed* iff *all its bunches are closed.* ∎

Therefore, any open tableau has at least one open bunch, formed by one or more open branches. To complete, this section we provide two examples: a closed tableau and an open tableau. We mark eventualities in gray boxes and use large circles to represent the generation of AND-nodes or bunches.

4.2 Examples

In this Subsection we provide some examples of systematic tableaux. For readability, the marked eventualities are in gray boxes. Big circles are used to represent AND-nodes or bunches. Whenever a bunch has a unique successor, we omit the big circle in the edge before the $(\mathsf{Q}\circ)$-successor.

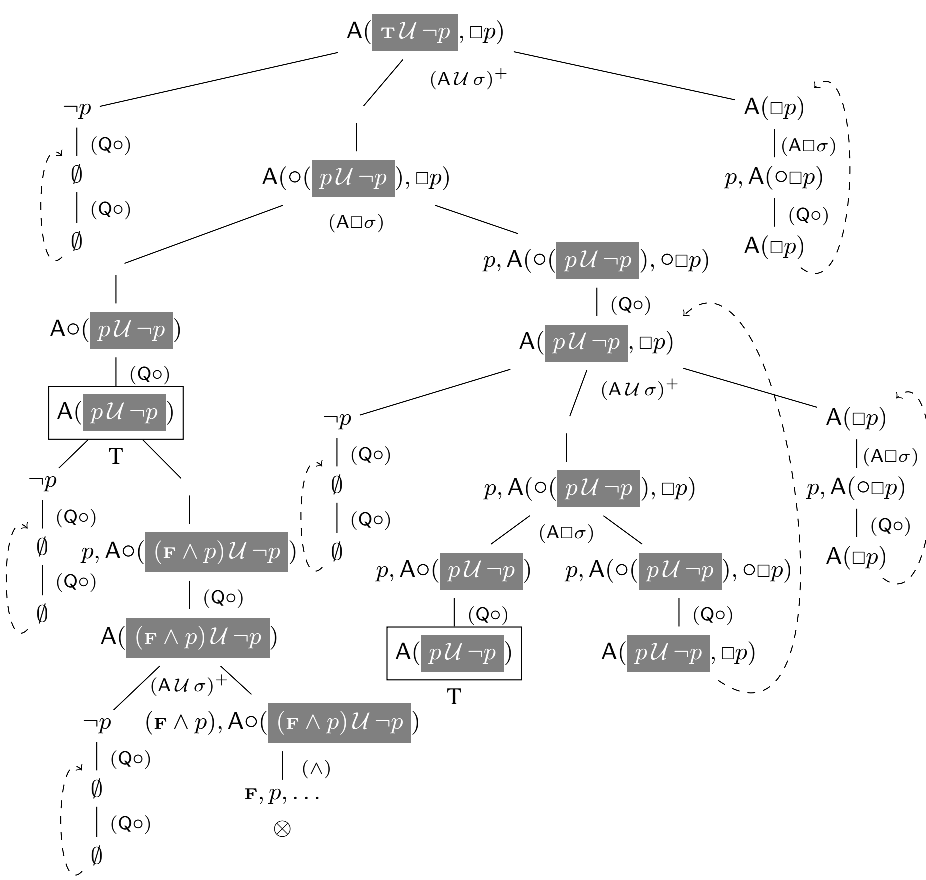

Fig. 11: An open tableau for $\mathsf{A}(\mathbf{T}\,\mathcal{U}\,\neg p, \square p)$

Example 30. In Figure 11 we depict an open tableaux for $\mathsf{A}(\mathbf{T}\,\mathcal{U}\,\neg p, \square p)$ or equivalently $\mathsf{A}(\diamondsuit\neg p, \square p)$. In fact, the tableau has only one closed branch –the fourth branch from the left. The remaining branches are finished by a loop-node. There are only three different labels of loop-nodes (that are repeated in different branches): the empty set of formulae, the singleton containing the potentially-cycling formula $\mathsf{A}(p\,\mathcal{U}\,\neg p, \Box p)$, and the singleton containing the cycling formula $\mathsf{A}\Box p$. ■

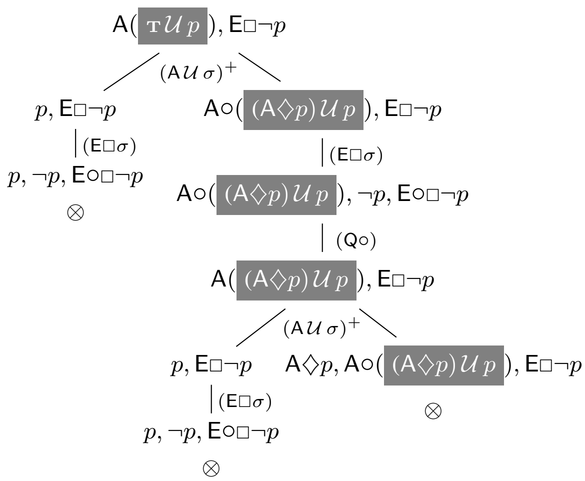

Fig. 12: A closed tableau for $\mathsf{A}(\mathbf{T}\,\mathcal{U}\,p), \mathsf{E}\Box\neg p$.

Example 31. In Figure 12 we depict a closed tableau for $\mathsf{A}(\mathbf{T}\,\mathcal{U}\,p), \mathsf{E}\Box\neg p$. Note that, in the two applications of the rule $(\mathsf{A}\,\mathcal{U}\,\sigma)^+$, the inner context is empty and the outer context is $\mathsf{E}\Box\neg p$ whose negation in nnf is $\mathsf{A}\Diamond p$. Henceforth, the rightmost leaf in Figure 12 is the result of simplifying the selected formula from $\mathsf{A}\circ(((\mathsf{A}\Diamond p) \wedge (\mathsf{A}\Diamond p))\,\mathcal{U}\,p)$ to $\mathsf{A}\circ((\mathsf{A}\Diamond p)\,\mathcal{U}\,p)$. This right-most branch is closed because $\mathsf{A}\Diamond p$ is the negation normal form of $\mathsf{E}\Box\neg p$. ■

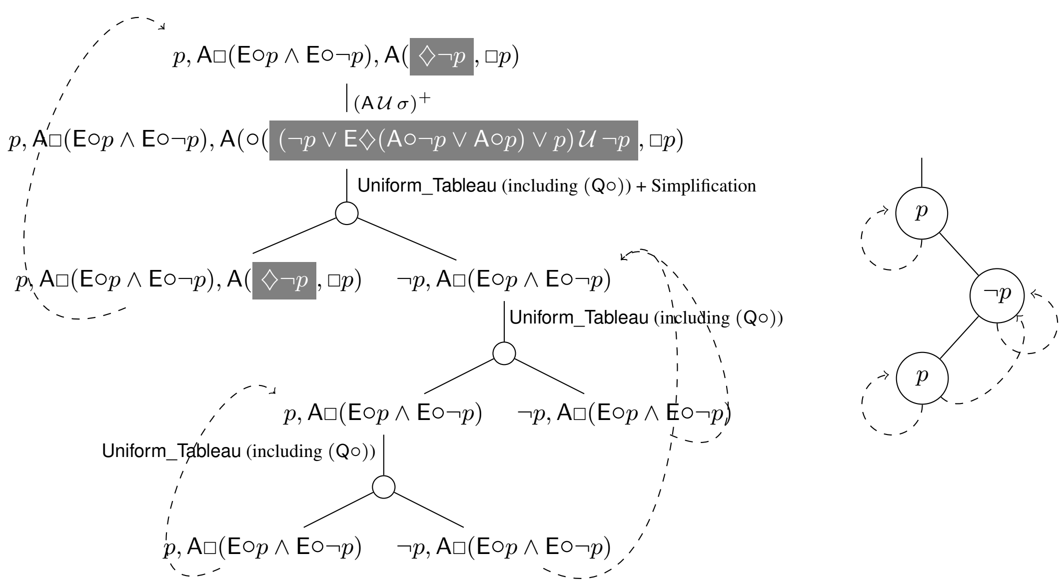

Fig. 13: Open bunch in the tableau for $p, \mathsf{A}\Box(\mathsf{E}\circ p \wedge \mathsf{E}\circ\neg p), \mathsf{A}(\Diamond\neg p, \Box p)$.

Example 32. On the left of Figure 13 we depict a representative open bunch of a tableau for the set of formulae:

$$\{p, \mathsf{A}\Box(\mathsf{E}\bigcirc p \wedge \mathsf{E}\bigcirc\neg p), \mathsf{A}(\Diamond\neg p, \Box p)\}.$$

In this example, we apply at once the Uniform_Tableau procedure subsequently choosing one of the leaves produced. Note that, for each node, we draw only one of the OR-children, but all the AND-children. The whole tableau, using the derived rules introduced in Section 5, is depicted in Figure 18, where all the rule applications are detailed. It is explained in Example 35.

Note that, for each node, we draw only one of the OR-children, but all the AND-children. In the marked eventuality, $\neg p \vee \mathsf{E}\Diamond(\mathsf{A}\circ\neg p)$ comes from the negation of the outer context, and the disjunct p from the inner context. By 'Simplification' $(\neg p \vee \mathsf{E}\Diamond(\mathsf{A}\circ\neg p \vee \mathsf{A}\circ p) \vee p)$ is reduced to $\mathbf{T}$ (in the left-hand children). In the right-hand node, $\neg p$ subsumes $\mathsf{A}((\dots)\,\mathcal{U}\,\neg p, \Box p)$. This open bunch represents the model (of the input set of formulae) that we depict in the right-hand of Figure 13. Since $\mathsf{A}(\Diamond\neg p, \Box p)$ is a valid formula, it is easy to see that it is a model of $p, \mathsf{A}\Box(\mathsf{E}\circ p \wedge \mathsf{E}\circ\neg p)$. ■

5 Derived Rules and Examples

In this section we introduce some derived rules and give some examples of tableaux.

α_Rule	α	S_α
$(\mathsf{A}\Diamond\Box)$	$\mathsf{A}(\Diamond\Box\sigma, \mathit{\Pi})$	$\{\sigma, \mathsf{A}(\circ\Diamond\Box\sigma, \mathit{\Pi})\}$
$(\mathsf{E}\Box\Diamond)$	$\mathsf{E}(\Box\Diamond\sigma), \mathit{\Pi})$	$\{\mathsf{E}(\Diamond\sigma, \circ\Box\Diamond\sigma, \mathit{\Pi})\}$
$(\mathsf{A}\Box\Diamond)$	$\mathsf{A}(\Box\Diamond\sigma, \mathit{\Pi})$	$\{\mathsf{A}(\Diamond\sigma, \mathit{\Pi}), \mathsf{A}(\circ\Box\Diamond\sigma, \mathit{\Pi})\}$

β_Rule	β	k	$S_{\beta_i} (1 \leq i \leq k)$
$(\mathsf{E}\Diamond\sigma)$	$\mathsf{E}(\Diamond(\sigma_1 \wedge \Diamond\sigma_2), \mathit{\Pi})$	2	$S_{\beta_1} = \{\sigma_1, \mathsf{E}(\Diamond\sigma_2, \mathit{\Pi}\}$ $S_{\beta_2} = \{\mathsf{E}(\circ\Diamond(\sigma_1 \wedge \Diamond\sigma_2), \mathit{\Pi})\}$
$(\mathsf{E}\Diamond\Box)$	$\mathsf{E}(\Diamond\Box\sigma, \mathit{\Pi})$	2	$S_{\beta_1} = \{\mathsf{E}(\Box\sigma, \mathit{\Pi})\}$ $S_{\beta_2} = \{\mathsf{E}(\circ\Diamond\Box\sigma, \mathit{\Pi})\}$
$(\mathsf{A}\Diamond\sigma)$	$\mathsf{A}(\Diamond(\sigma_1 \wedge \Diamond\sigma_2), \mathit{\Pi})$	3	$S_{\beta_1} = \{\sigma_1, \mathsf{A}(\Diamond\sigma_2, \mathit{\Pi})\}$ $S_{\beta_2} = \{\mathsf{A}(\circ\Diamond(\sigma_1 \wedge \Diamond\sigma_2), \mathit{\Pi})\}$ $S_{\beta_3} = \{\mathsf{A}\mathit{\Pi}\}$

β^+-Rule	$\mathit{\Sigma}, \beta$	k	$S^+_{\mathit{\Sigma},\beta_i} (1 \leq i \leq k)$
$(\mathsf{A}\Diamond\sigma)^+$	$\mathit{\Sigma}, \mathsf{A}(\Diamond(\sigma_1 \wedge \Diamond\sigma_2), \mathit{\Pi})$	3	$S^+_{\mathit{\Sigma},\beta_1} = \{\sigma_1, \mathsf{A}(\Diamond\sigma_2, \mathit{\Pi})\}$ $S^+_{\mathit{\Sigma},\beta_2} = \{\mathsf{A}(\circ((\neg\mathit{\Sigma} \vee \varphi_{\mathit{\Pi}})\,\mathcal{U}\,(\sigma_1 \wedge \Diamond\sigma_2)), \mathit{\Pi})\}$ $S^+_{\mathit{\Sigma},\beta_3} = \{\mathsf{A}\mathit{\Pi}\}$
$(\mathsf{A}\Diamond\Box)^+$	$\mathit{\Sigma}, \mathsf{A}(\Diamond\Box\sigma, \mathit{\Pi})$	2	$S^+_{\mathit{\Sigma},\beta_1} = \{\mathsf{A}(\Box\sigma, \mathit{\Pi})\}$ $S^+_{\mathit{\Sigma},\beta_2} = \{\mathsf{A}(\circ((\neg\mathit{\Sigma} \vee \varphi_{\mathit{\Pi}} \vee \sigma)\,\mathcal{U}\,\Box\sigma), \mathit{\Pi})\}$

β^+-Rule	$\mathit{\Sigma}, \beta$	k	$S^+_{\mathit{\Sigma},\beta_i} (1 \leq i \leq k)$
$(\mathsf{E}\Diamond\sigma\Box)^+$	$\mathit{\Sigma}, \mathsf{E}\mathit{\Pi}$	$2n$	$S^+_{\mathit{\Sigma},\beta_1}$ $\vdots$ $S^+_{\mathit{\Sigma},\beta_i} \begin{cases} \{\sigma_1, \mathsf{E}(\Diamond\sigma_2, \mathit{\Pi}^{-i})\} & \text{if } \pi_i = \Diamond(\sigma_1 \wedge \Diamond\sigma_2) \\ \{\mathsf{E}(\Box\sigma, \mathit{\Pi}^{-i})\} & \text{if } \pi_i = \Diamond\Box\sigma \end{cases}$ $\vdots$ $S^+_{\mathit{\Sigma},\beta_n}$ $S^+_{\mathit{\Sigma},\beta_{n+1}}$ $\vdots$ $S^+_{\mathit{\Sigma},\beta_{2i}} \begin{cases} \{\mathsf{E}(\circ((\neg\mathit{\Sigma})\,\mathcal{U}\,(\sigma_1 \wedge \Diamond\sigma_2)), \mathit{\Pi}^{-i})\} & \text{if } \pi_i = \Diamond(\sigma_1 \wedge \Diamond\sigma_2) \\ \{\mathsf{E}(\circ((\neg\mathit{\Sigma})\,\mathcal{U}\,\Box\sigma), \mathit{\Pi}^{-i})\} & \text{if } \pi_i = \Diamond\Box\sigma \end{cases}$ $\vdots$ $S^+_{\mathit{\Sigma},\beta_{2n}}$

Fig. 14: DERIVED RULES. (Notation: σ, σ_i stand for state formulae, $\mathit{\Sigma}$ is a (possibly empty) set of state formulae, $\mathit{\Pi}$ is a (possibly empty) set of basic-path formulae. Formula $\varphi_{\mathit{\Pi}}$ is defined in Def. 19. We denote by $\mathit{\Pi}_{\mathcal{U}}$ the set of all formulae in $\mathit{\Pi}$ that have the forms $\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$ and $\sigma_1\,\mathcal{U}\,\Box\sigma_2$. $\mathit{\Pi}_{\mathcal{U}}$ is enumerated as $\{\pi_1, \ldots, \pi_n\}$ for $n \geq 1$, and $\mathit{\Pi}^{-i} = \mathit{\Pi} \setminus \{\pi_i\}$.)

We introduce some trivially correct rules that just allow us to make tableaux shorter and easier to understand.

First, we introduce a refinement of the rules β^+-rules. It is easy to see, that instead of $\neg\Sigma$ in these rules we can write $\neg\Sigma'$ where

$$\Sigma' = \Sigma \setminus \{\mathsf{A}(\pi_1, \ldots, \pi_n) \mid \mathsf{A}(\pi_1, \ldots, \pi_n) \in \Sigma \text{ and for all } i : \pi_i \text{ is } \Box\varphi \text{ or } \circ\Box\varphi\}.$$

This refinement just prevents to generate (at each application of a β^+-rule) one closed branch for each formula of the form $\mathsf{A}(\pi_1, \ldots, \pi_n)$ that belongs to Σ. The idea is that the formula

$$A([\circ]\Box\varphi_1, \ldots, [\circ]\Box\varphi_n)$$

is invariantly preserved from one stage to another (because of the rules $(\mathsf{A}\Box\sigma)$, $(\mathsf{A}\Box\mathcal{U})$ and $(\mathsf{Q}\circ)$), so the next application of the beta-plus rule to the selected eventuality produce a branch containing

$$\neg\mathsf{A}([\circ]\Box\varphi_1, \ldots, [\circ]\Box\varphi_n).$$

However, this branch is closed at once because it also contains $\mathsf{A}([\circ]\Box\varphi_1, \ldots, [\circ]\Box\varphi_n)$. Interestingly, the formulae of the form $\mathsf{E}(\Box\varphi_1, \ldots, \Box\varphi_n)$ has not the same property. Consequently, in this section, any application of a β^+-rule will get $\neg\Sigma'$ instead of $\neg\Sigma$. Of course, for empty Σ there is no difference, we get $\mathbf{F}$ in both options.

Second, in Figure 14, we introduce the derived rules for the temporal operator $\Diamond$, using its definition in terms of $\mathcal{U}$: $\Diamond\varphi \equiv \mathbf{T}\,\mathcal{U}\,\varphi$.

Third, subsumption is also very useful to make shorter (but equivalent) nodes in tableaux. In the examples we will develop in this section, we use the following subsumption-based equivalences along with the simplification rules of Subsection 3.5. We use them just in the application of the rule $(\mathsf{Q}\circ)$ (implicitly), when possible. Derived from the simplification rules in Subsection 3.5:

$(\sqsubset\mathsf{E}\Box\Diamond)$ $\mathsf{E}(\Diamond\sigma, \Box\Diamond\sigma), \Pi) \equiv \mathsf{E}(\Box\Diamond\sigma, \Pi)$.
$(\sqsubset\mathsf{A}\Box\Diamond)$ If $\Pi' \subseteq \Pi$ then $\mathsf{A}(\Diamond\sigma', \Pi) \wedge \mathsf{A}(\Box\Diamond\sigma', \Pi') \equiv \mathsf{A}(\Box\Diamond\sigma', \Pi')$.
$(\sqsubset\mathsf{A}\sigma\Diamond)$ $\sigma \wedge \mathsf{A}(\Diamond\sigma, \Pi) \equiv \sigma$.
$(\sqsubset\mathsf{E}\Diamond\sigma)$ $\mathsf{E}(\sigma\,\mathcal{U}\,\varphi, \Diamond\varphi, \Pi) \equiv \mathsf{E}(\sigma\,\mathcal{U}\,\varphi, \Pi)$
$(\sqsubset\mathsf{A}\Diamond\sigma)$ If $\Pi' \subseteq \Pi$ then $\mathsf{A}(\sigma\,\mathcal{U}\,\varphi, \Pi') \wedge \mathsf{A}(\Diamond\varphi, \Pi) \equiv \mathsf{A}(\sigma\,\mathcal{U}\,\varphi, \Pi')$.

Other useful simplification rules are:

$(\sqsubset\mathsf{A}\Pi)$ $\mathsf{A}\Pi \wedge \mathsf{A}\Pi' \equiv \mathsf{A}\Pi'$ if $\Pi' \subseteq \Pi$.
$(\sqsubset\mathsf{E}\Pi)$ $\mathsf{E}\Pi \wedge \mathsf{E}\Pi' \equiv \mathsf{E}\Pi$ if $\Pi' \subseteq \Pi$.

In the rest of this section we provide some examples of systematic tableau constructions. They are explained in Examples 33, 34 and 35. As in the previous examples, the marked eventualities are in gray boxes; big circles are used to represent AND-nodes or bunches; and we omit the big circle in the edge before the $(\mathsf{Q}\circ)$-successor,whenever a bunch has a unique successor.

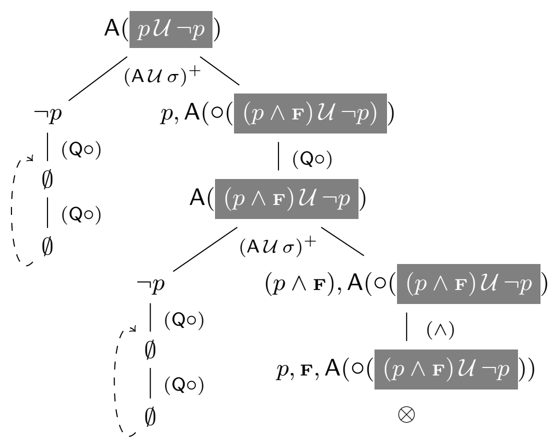

Fig. 15: An open tableau for $\mathsf{A}(p\,\mathcal{U}\,\neg p)$.

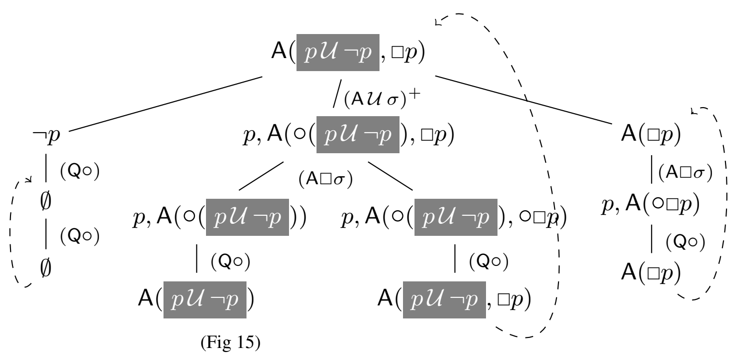

Fig. 16: An open tableau for $\mathsf{A}(p\,\mathcal{U}\,\neg p, \Box p)$

Example 33. In Figure 15 we depict an open tableau for the singleton $\{\mathsf{A}(p\,\mathcal{U}\,\neg p)\}$ that is a sub-tableau of the tableau in Figure 16. In the first tableau, both the outer and the inner context are empty, for the marked eventuality. However, in Figure 16, the eventuality marked in the root has empty outer context, but the inner context is $\Box p$, hence the formula in the left-hand side of the $\mathcal{U}$ is p as simplification of $\textbf{T} \wedge (\textbf{F} \vee p)$. Note that the central open branch represents a model of $\Box p$. ■

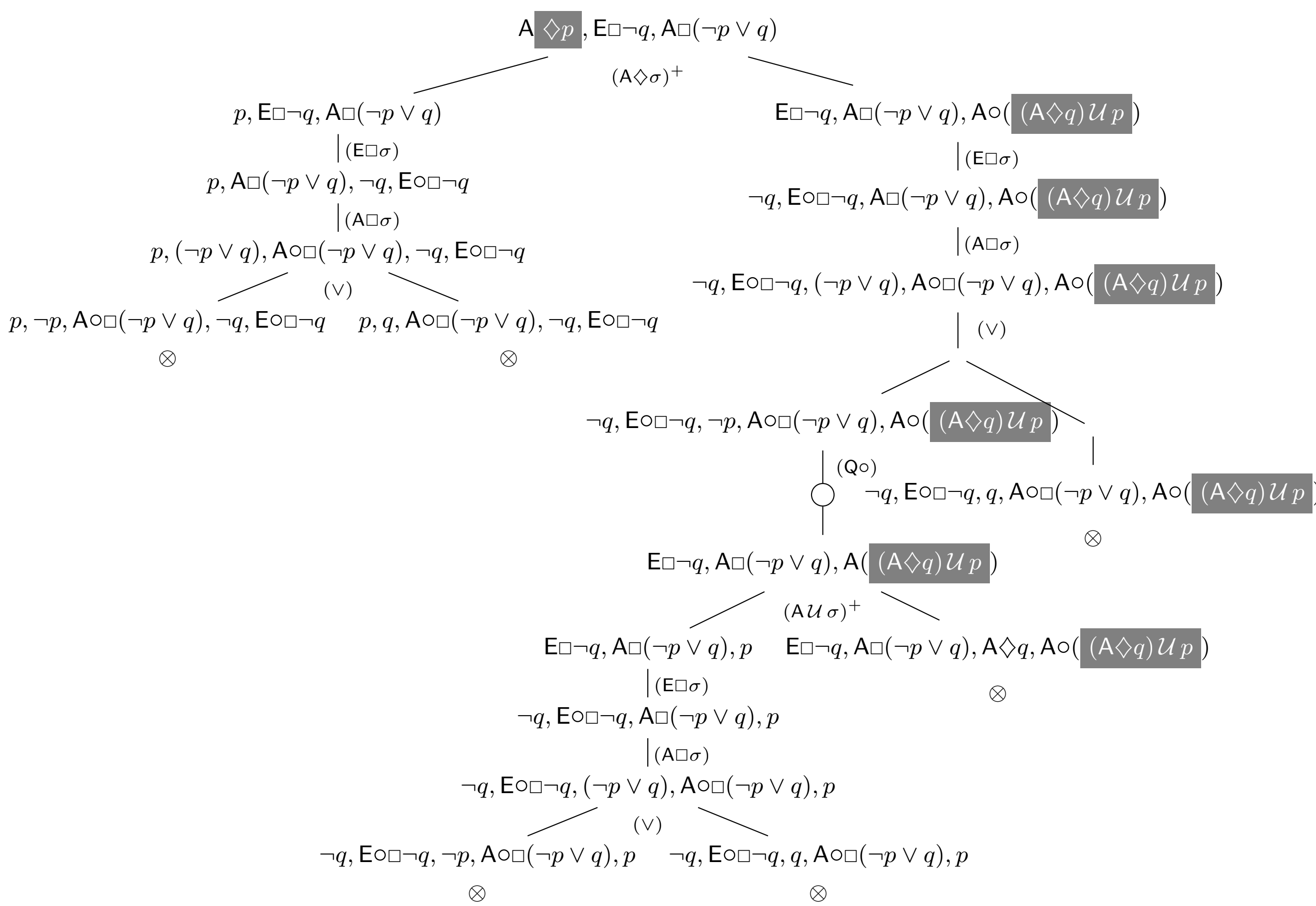

Fig. 17: A closed tableau for $\mathsf{A}\Diamond p, \mathsf{E}\Box\neg q, \mathsf{A}\Box(\neg p \vee q)$.

Example 34. In Figure 17 we depict a closed tableau that proves the unsatisfiability of the set of formulae: $\mathsf{A}\Diamond p, \mathsf{E}\Box\neg q, \mathsf{A}\Box(\neg p \vee q)$. For the selected eventuality the inner context is empty, but the outer context is formed by two formulas: $\mathsf{E}\Box\neg q$ whose negation in normal form is $\mathsf{A}\Diamond q$ and the formula $A\Box(\neg p \vee q)$ which, as explained above, is excluded from the negation of the context, because the branch it generates is necessarily closed. Note that the unique application of $(\mathsf{Q}\circ)$ enlarge the branch with a unique child for the only E-formula. ∎

Example 35. In Figure 18 we show the open tableau for $p, \mathsf{A}\Box(\mathsf{E}\circ p \wedge \mathsf{E}\circ\neg p), \mathsf{A}(\Diamond\neg p, \Box p)$.
We have named as $C1$ and $C2$ two sub-tableaux that are repeated.
For the eventuality marked in the root none of both context is empty, thought the formula $\mathsf{A}\Box(\mathsf{E}\circ p \wedge \mathsf{E}\circ\neg p)$ is excluded in the negated outer context, as explained above. Though the formula $\neg p \vee p$ could be simplified as $\mathbf{T}$, however we keep it unsimplified in this example for illustrating that not all simplification are crucial for termination. In fact, this is not, though many trivial simplifications are very useful for efficient.
Note that the left-most bunch (of two branches) is closed because its left-most branch is closed, though its right branch is open.
One of the open bunches of this tableau (and the model its represents) is given in Figure 13 and explained in Example 32. ∎

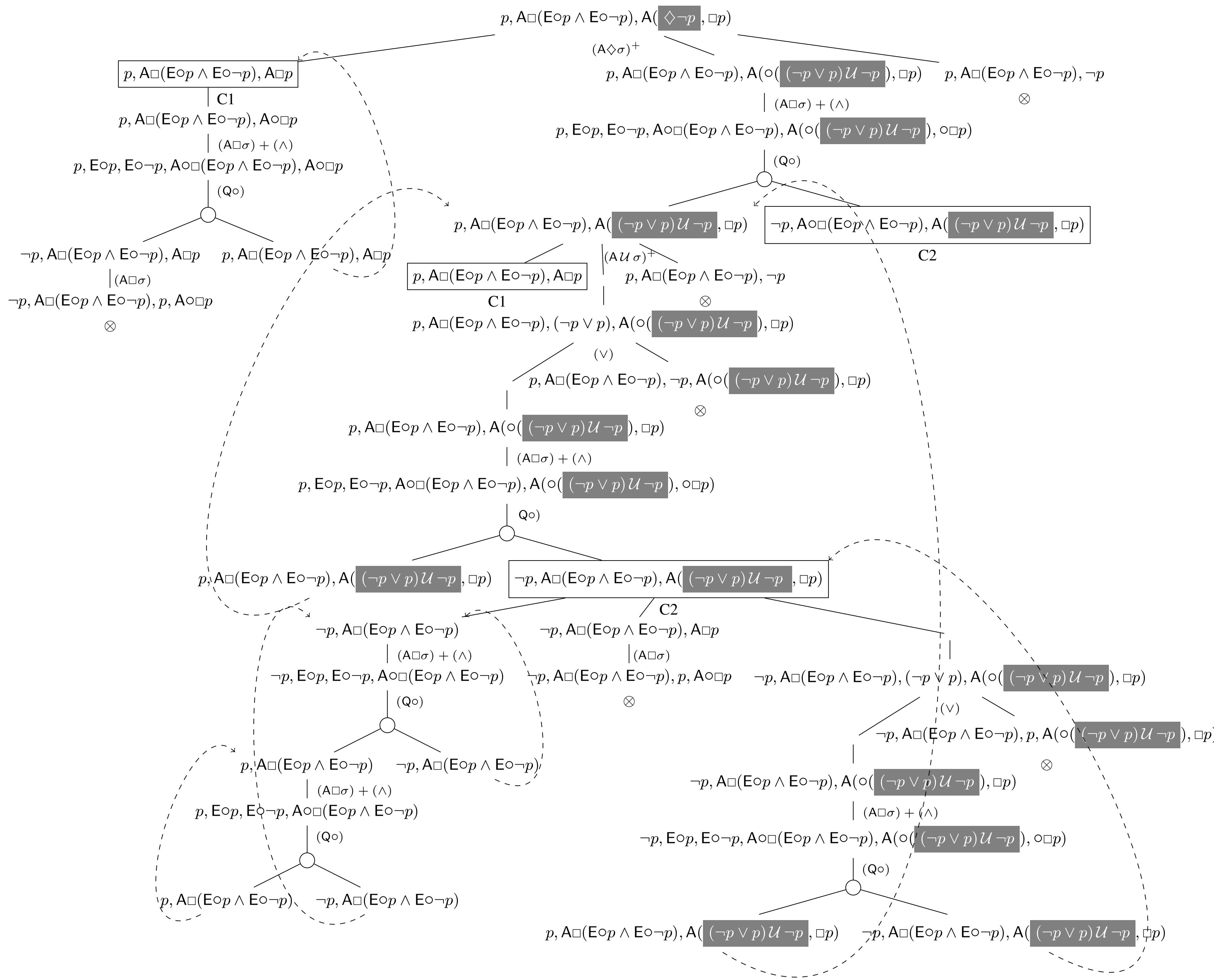

Fig. 18: An open tableau for $p, \mathsf{A}\Box(\mathsf{E}\circ p \wedge \mathsf{E}\circ\neg p), \mathsf{A}(\Diamond\neg p, \Box p)$.

6 Soundness

To prove the soundness of our tableau method (Theorem 38) we show that every tableau rule preserves satisfiability (Lemma 37). To prove the latter we essentially use the limit closure property, ensuring that the satisfiability of the negated inner context is preserved from segments of a limit path to the limit path itself (Proposition 36). The use of φ_Π (Definition 19) is crucial here.

Proposition 36 (Preservation of the Negated Inner Context). *Let Π be any set of basic path formulae and let φ_Π be as in Definition 19. Let $\mathcal{K}$ be a Kripke structure, $x_1 \in \mathsf{fullpaths}(\mathcal{K})$ such that $\mathcal{K}, x1, 0 \models \neg\Pi$. Let $y = x_1^{\leq i_1} x_2^{\leq i_2} \cdots x_k^{\leq i_k} \cdots$ be a limit path in $\mathsf{fullpaths}(\mathcal{K})$, for some $i_1 > 0$ and some $x_2^{\leq i_2} \cdots x_k^{\leq i_k} \cdots$. Then $\mathcal{K}, y, 0 \models \neg\pi$ holds for all $\pi \in \Pi$, provided that the following two conditions hold for all $n \geq 1$:*

(a) $\mathcal{K}, x_1^{\leq i_1} x_2^{\leq i_2} \cdots x_n, j \models \neg\sigma_2$ *for all* $\sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3) \in \Pi$ *and all* $j \in \{0..i_n\}$*, and*
(b) $\mathcal{K}, x_1^{\leq i_1} x_2^{\leq i_2} \cdots x_n^{\leq i_n}, i_n \models \neg\varphi_\Pi$.

Proof. We check the four cases of a basic path formula $\pi \in \Pi$. If π is of the form $\circ\sigma$, then $\mathcal{K}, y, 0 \models \neg\circ\sigma$ because $\mathcal{K}, y, 0 \models \neg\pi$ and $i_1 > 0$. If π is of the form $\sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3)$, then property (a) ensures that every state in y satisfies $\neg\sigma_2$. Therefore, $\neg(\sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3))$ is satisfied along the limit path y. For the remaining three cases, on the basis of (b) and Definition 19, we can prove the following three facts: (1) If $\pi = \Box(\sigma_1 \vee \Box\sigma_2)$, then $\mathcal{K}, x_1^{\leq i_1} x_2^{\leq i_2} \cdots x_n^{\leq i_n}, i_n \models \neg\sigma_1 \wedge \neg\sigma_2$ for all n. (2) If $\pi = \Box(\sigma_1 \mathcal{U} \sigma_2)$, then we have that $\mathcal{K}, x_1^{\leq i_1} x_2^{\leq i_2} \cdots x_n^{\leq i_n}, i_n \models \neg\mathsf{E}(\Diamond\sigma_2)$ for all n. (3) If $\pi = \sigma_1 \mathcal{U} \Box\sigma_2$, then $\mathcal{K}, x_1^{\leq i_1} x_2^{\leq i_2} \cdots x_n^{\leq i_n}, i_n \models \neg\sigma_2$ for all n. Therefore, in any of the three cases, we can ensure that $\mathcal{K}, y, 0 \models \pi$. ∎

Lemma 37 (Soundness of the Tableau Rules). *For any set of state formulae Σ:*

(i) For any α-formula α : $\mathsf{Sat}(\Sigma, \alpha)$ iff $\mathsf{Sat}(\Sigma, S_\alpha)$.
(ii) For any β-formula β of range k: $\mathsf{Sat}(\Sigma, \beta)$ iff $\mathsf{Sat}(\Sigma, S_{\beta_i})$ *for some* $1 \leq i \leq k$.
(iii) For any β^+-formula β of range k: $\mathsf{Sat}(\Sigma, \beta)$ iff $\mathsf{Sat}(\Sigma, S^+_{\Sigma,\beta_i})$ *for some* $1 \leq i \leq k$.
(iv) If Σ is a set of consistent literals: $\mathsf{Sat}(\Sigma, \mathsf{A}\circ\Phi_1, \ldots, \mathsf{A}\circ\Phi_n, \mathsf{E}\circ\Psi_1, \ldots, \mathsf{E}\circ\Psi_m)$ iff *for all* $0 \leq i \leq m$: $\mathsf{Sat}(\mathsf{A}\Phi_1, \ldots, \mathsf{A}\Phi_n, \mathsf{E}\Psi_i)$. ∎

Proof. Noting that (i), (ii) and (iv) can be easily proved by the 'systematic' application of the semantic definitions of temporal operators, we prove (iii). The 'only if' direction' for each of the cases of β^+-rules is trivial. We will prove the 'if' direction of the three rules $(\mathsf{E}\mathcal{U}\sigma\Box)^+$, $(\mathsf{A}\mathcal{U}\sigma)^+$ and $(\mathsf{A}\mathcal{U}\Box)^+$, in this order.
For the 'if' direction of rule $(\mathsf{E}\mathcal{U}\sigma\Box)^+$, let us suppose that $\mathcal{K} \models \Sigma, \mathsf{E}\Pi$, where Π contains at least one eventuality. There exists $x \in \mathsf{fullpaths}(\mathcal{K})$ such that $\mathcal{K}, x, 0 \models \Sigma, \Pi$. We are going to prove that there exists $\mathcal{K}'$ such that one of the following properties holds:

(a) $\mathcal{K}' \models \Sigma, \sigma_2, \mathsf{E}(\Diamond\sigma_3, \Pi^{-i})$ for some $\pi_i = \sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3)$ in $\Pi_\mathcal{U}$
(b) $\mathcal{K}' \models \Sigma, \mathsf{E}(\Box\sigma_2, \Pi^{-i})$ for some $\pi_i = \sigma_1 \mathcal{U} \Box\sigma_2$ in $\Pi_\mathcal{U}$
(c) $\mathcal{K}' \models \Sigma, \sigma_1, \mathsf{E}(\circ((\sigma_1 \wedge \neg\Sigma) \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3)), \Pi^{-i})$ for some $\pi_i = \sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3)$ in $\Pi_\mathcal{U}$
(d) $\mathcal{K}' \models \Sigma, \sigma_1, \mathsf{E}(\circ((\sigma_1 \wedge \neg\Sigma) \mathcal{U} \Box\sigma_2), \Pi^{-i})$ for some $\pi_i = \sigma_1 \mathcal{U} \Box\sigma_2$ in $\Pi_\mathcal{U}$.

Since $\mathcal{K}, x, 0 \models \pi_i$ for all $\pi \in \Pi_\mathcal{U}$, we define j to be the least $i \geq 0$ such that $\mathcal{K}, x, i \models \varphi$ for some $\sigma_1 \mathcal{U} \varphi \in \Pi_\mathcal{U}$. If $j = 0$, then (a) and (b) (depending on the form of φ) are trivially satisfied for $\mathcal{K}' = \mathcal{K}$. Otherwise, if $j > 0$, it holds that $\mathcal{K}, x, j \models \varphi$ and for all $i < j$: $\mathcal{K}, x, i \models \sigma_1$. Moreover, as j is the minimal index, for all $0 \leq i < j$: $\mathcal{K}, x, i \models \sigma_1'$ for all $\sigma_1' \mathcal{U} \varphi' \in \Pi_\mathcal{U}$. Consider k to be the greatest index i such that $0 \leq i < j$ and $\mathcal{K}, x, i \models \Sigma$. Henceforth, we have that $\mathcal{K}, x, k \models \Sigma$ and $\mathcal{K}, x, h \models \neg\Sigma$, for all h such that $k + 1 \leq h < j$. Therefore, (c) and (d) hold for $\mathcal{K}' = \mathcal{K} \upharpoonright x(k)$.

For the 'if' direction of the rule $(\mathsf{A}\mathcal{U}\sigma)^+$, let us suppose that the three sets $\Sigma \cup S_{\Sigma,\beta_1}$, $\Sigma \cup S_{\Sigma,\beta_2}$, and $\Sigma \cup S_{\Sigma,\beta_3}$ of the rule $(\mathsf{A}\mathcal{U}\sigma)^+$ are unsatisfiable. We will show that the set $\Sigma, \mathsf{A}(\sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3), \Pi)$ must be also unsatisfiable. By the hypothesis, we know that any model of Σ is not a model of S_{Σ,β_i} for all $i \in \{1, 2, 3\}$. In other words, for any $\mathcal{K}$ such that $\mathcal{K} \models \Sigma$, the followings three facts holds:

(a) $\mathcal{K} \not\models \sigma_2 \wedge \mathsf{A}(\Diamond\sigma_3, \Pi)$

(b) $\mathcal{K} \not\models \sigma_1 \wedge \mathsf{A}(\circ((\sigma_1 \wedge (\neg\Sigma \vee \varphi_\Pi))\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)), \Pi)$
(c) $\mathcal{K} \not\models \mathsf{A}\Pi$

To show that $\Sigma, \mathsf{A}(\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3), \Pi)$ is unsatisfiable, we consider an arbitrary $\mathcal{K}$ such that $\mathcal{K} \models \Sigma$ and prove that $\mathcal{K} \not\models \mathsf{A}(\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3), \Pi)$. Since $\mathcal{K} \models \Sigma$, then (a), (b) and (c) hold. According to (b), there are two possible cases:
(Case 1): If $\mathcal{K} \not\models \sigma_1$ then, by (a), either $\mathcal{K} \models \neg\sigma_1 \wedge \neg\sigma_2$ or $\mathcal{K} \models \neg\sigma_1 \wedge \mathsf{E}(\Box\neg\sigma_3, \neg\Pi)$. In both cases, it is easy to see that $\mathcal{K} \not\models \mathsf{A}(\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3), \Pi)$.
(Case 2): Otherwise, if $\mathcal{K} \not\models \mathsf{A}(\circ((\sigma_1 \wedge (\neg\Sigma \vee \varphi_\Pi))\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)), \Pi)$, then there exists $x_1 \in \mathsf{fullpaths}(\mathcal{K})$ such that $\mathcal{K}, x_1, 0 \models \circ\neg((\sigma_1 \wedge (\neg\Sigma \vee \varphi_\Pi))\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)) \wedge \neg\Pi$. This yields two possible cases:
(Case 2.1): If $\mathcal{K}, x_1, 0 \models \circ\Box(\neg\sigma_2 \vee \Box\neg\sigma_3) \wedge \neg\Pi$, then it is trivial that $\mathcal{K} \not\models \mathsf{A}(\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3), \Pi)$.
(Case 2.2): Otherwise, there should exist $i_1 > 0$ that satisfies the following three properties:

(i) $\mathcal{K}, x_1, j \models (\neg\sigma_2) \vee \Box\neg\sigma_3$ for all j such that $0 \leq j \leq i_1$, and
(ii) $\mathcal{K}, x_1, i_1 \models \neg\sigma_1 \vee (\Sigma \wedge \neg\varphi_\Pi)$, and
(iii) $\mathcal{K}, x_1, 0 \models \neg\Pi$

If (i) is satisfied because $\mathcal{K}, x_1, j \models \Box\neg\sigma_3$ (for some j such that $0 \leq j \leq i_1$) then trivially $\mathcal{K}, x_1, 0 \not\models \sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$. This, along with the fact (iii), not only ensures that $\mathcal{K} \not\models \mathsf{A}(\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3), \Pi)$ but also applies to any other formula $\sigma_1'\,\mathcal{U}\,(\sigma_2' \wedge \Diamond\sigma_3')$ in Π. Henceforth, in what follows, we can suppose that for all j such that $0 \leq j \leq i_1$: $\mathcal{K}, x_1, j \models \neg\sigma_2$ and also that $\mathcal{K}, x_1, j \models \neg\sigma_2'$ for all $\sigma_1'\,\mathcal{U}\,(\sigma_2' \wedge \Diamond\sigma_3') \in \Pi$. If (ii) is satisfied because $\mathcal{K}, x_1, i_1 \models \neg\sigma_1$ then it is clear that $\mathcal{K}, x_1, 0 \not\models \sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$. Therefore, by (i) and (iii), $\mathcal{K} \not\models \mathsf{A}(\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3), \Pi)$.
Otherwise, if (ii) is satisfied because $\mathcal{K}, x_1, i_1 \models \Sigma \wedge \neg\varphi_\Pi$, then (a), (b) and (c) hold for $\mathcal{K} \upharpoonright x_1(i_1)$ (instead of $\mathcal{K}$) because $\mathcal{K}, x_1, i_1 \models \Sigma$. Hence, applying the same reasoning for $\mathcal{K} \upharpoonright x_1(i_1)$ as we did above for $\mathcal{K}$, we conclude that there should be a path $x_2 \in \mathsf{fullpaths}(\mathcal{K} \upharpoonright x_1(i_1))$ such that one of the following two facts holds:
(Case 2.2.1): $\mathcal{K} \upharpoonright x_1(i_1), x_2 \models \Box\neg(\sigma_2 \wedge \Diamond\sigma_3) \wedge \neg\Pi$, and therefore $\mathcal{K} \not\models \mathsf{A}(\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3), \Pi)$.
(Case 2.2.2): there should exist $i_2 > 0$ such that $\mathcal{K} \upharpoonright x_1(i_1), x_2, i_2 \models \Sigma \wedge \neg\varphi_\Pi$ and for all $j \in \{0..i_2\}$:

- $\mathcal{K} \upharpoonright x_1(i_1), x_2, j \models \neg\sigma_2$, and
- $\mathcal{K} \upharpoonright x_1(i_1), x_2, j \models \neg\sigma_2'$ for all $\sigma_1'\,\mathcal{U}\,(\sigma_2' \wedge \Diamond\sigma_3') \in \Pi$

Now, (a), (b) and (c) apply to $\mathcal{K} \upharpoonright x_2(i_2)$. Hence, the infinite iteration of the second case yields a path $y = x_1^{\leq i_1} x_2^{\leq i_2} \cdots x_k^{\leq i_k} \cdots$ (that exists by the limit closure property) for which the Proposition 36 ensures that $\mathcal{K}, y, 0 \not\models \mathsf{A}(\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3), \Pi)$.

The proof for the $(\mathsf{A}\mathcal{U}\Box)^+$ rule follows the same scheme. Let us suppose that the two sets $\Sigma \cup S_{\Sigma,\beta_1}$ and $\Sigma \cup S_{\Sigma,\beta_2}$, of the rule $(\mathsf{A}\mathcal{U}\Box)^+$ are unsatisfiable. We will show that the set $\Sigma, \mathsf{A}(\sigma_1\,\mathcal{U}\,\Box\sigma_2, \Pi)$ must be also unsatisfiable. By the hypothesis, we know that any model of Σ is not a model of S_{Σ,β_i} for all $i \in \{1,2\}$. In other words, for any $\mathcal{K}$ such that $\mathcal{K} \models \Sigma$, the followings two facts holds:

(a) $\mathcal{K} \not\models \mathsf{A}(\Box\sigma_2, \Pi)$
(b) $\mathcal{K} \not\models \sigma_1 \wedge \mathsf{A}(\circ((\sigma_1 \wedge (\neg\Sigma \vee \varphi_\Pi \vee \sigma_2))\,\mathcal{U}\,\Box\sigma_2), \Pi)$

To show that $\Sigma, \mathsf{A}(\sigma_1\,\mathcal{U}\,\Box\sigma_2, \Pi)$ is unsatisfiable, we consider an arbitrary $\mathcal{K}$ such that $\mathcal{K} \models \Sigma$ and prove that $\mathcal{K} \not\models \mathsf{A}(\sigma_1\,\mathcal{U}\,\Box\sigma_2, \Pi)$. Since $\mathcal{K} \models \Sigma$, then both (a) and (b) hold. According to (b), there are two possible cases:
(Case 1): If $\mathcal{K} \not\models \sigma_1$ then, by (a), $\mathcal{K} \models \neg\sigma_1 \wedge \mathsf{E}(\Diamond\neg\sigma_2, \neg\Pi)$. In this case, it is easy to see that $\mathcal{K} \not\models \mathsf{A}(\sigma_1\,\mathcal{U}\,\Box\sigma_2, \Pi)$.
(Case 2): Otherwise, if $\mathcal{K} \not\models \mathsf{A}(\circ((\sigma_1 \wedge (\neg\Sigma \vee \varphi_\Pi \vee \sigma_2))\,\mathcal{U}\,\Box\sigma_2), \Pi)$, then there exists $x_1 \in \mathsf{fullpaths}(\mathcal{K})$ such that $\mathcal{K}, x_1, 0 \models \circ\neg((\sigma_1 \wedge (\neg\Sigma \vee \varphi_\Pi \vee \sigma_2))\,\mathcal{U}\,\Box\sigma_2) \wedge \neg\Pi$. That is, there should exist $i_1 > 0$ that satisfies the following three properties:

(i) $\mathcal{K}, x_1, j \models \Diamond\neg\sigma_2$ for all j such that $0 \leq j \leq i_1$, and
(ii) $\mathcal{K}, x_1, i_1 \models \neg\sigma_1 \vee (\Sigma \wedge \neg\varphi_\Pi \wedge \neg\sigma_2)$, and
(iii) $\mathcal{K}, x_1, 0 \models \neg\Pi$

If (ii) is satisfied because $\mathcal{K}, x_1, i_1 \models \neg\sigma_1$ then trivially $\mathcal{K}, x_1, 0 \not\models \sigma_1 \,\mathcal{U}\, \Box\sigma_2$ as $\mathcal{K}, x_1, i_1 \models \Diamond\neg\sigma_2$ by item (i). Otherwise, if (ii) is satisfied because $\mathcal{K}, x_1, i_1 \models \Sigma \wedge \neg\varphi_\Pi \wedge \neg\sigma_2$, then (a) and (b) hold for $\mathcal{K} \restriction x_1(i_1)$ (instead of $\mathcal{K}$) because $\mathcal{K}, x_1, i_1 \models \Sigma$. Hence, applying the same reasoning for $\mathcal{K} \restriction x_1(i_1)$ as we did above for $\mathcal{K}$, we conclude that there should be a path $x_2 \in \mathsf{fullpaths}(\mathcal{K} \restriction x_1(i_1))$ such that one of the following two facts holds:

(Case 2.2.1): $\mathcal{K} \restriction x_1(i_1), x_2 \models \neg\sigma_1$, and therefore $\mathcal{K} \not\models \mathsf{A}(\sigma_1 \,\mathcal{U}\, \Box\sigma_2, \Pi)$.

(Case 2.2.2): there should exist $i_2 > 0$ such that $\mathcal{K} \restriction x_1(i_1), x_2, i_2 \models \Sigma \wedge \neg\varphi_\Pi \wedge \neg\sigma_2$ and for all $j \in \{0..i_2\}$ it holds that $\mathcal{K} \restriction x_1(i_1), x_2, j \models \Diamond\neg\sigma_2$. Now, (a), (b) and (c) apply to $\mathcal{K} \restriction x_2(i_2)$. Hence, the infinite iteration of the second case yields a path $y = x_1^{\leq i_1} x_2^{\leq i_2} \cdots x_k^{\leq i_k} \cdots$ (that exists by the limit closure property) for which the Proposition 32 ensures that $\mathcal{K}, y, 0 \not\models \mathsf{A}(\sigma_1 \,\mathcal{U}\, \Box\sigma_2, \Pi)$. ∎

Theorem 38 (Soundness of the Tableau Method). *Given any set of state formulae Σ, if there exists a closed tableau for Σ then* $\mathsf{UnSat}(\Sigma)$.

Proof. Let T_Σ be a closed tableau for Σ. The set of formulae labelling at least one leaf in each bunch is inconsistent and therefore unsatisfiable. Then, by Lemma 37, the labelling of the root node, Σ, is unsatisfiable. ∎

7 Completeness

In this section, we prove the completeness of the introduced tableau method. First, we define the notions of stage, expanded bunch, and expanded tableau. Then we prove the refutational completeness: every unsatisfiable set of state formulae has a closed tableau. In fact, we are going to prove that, for any set Σ_0, if the systematic tableau $\mathcal{A}^{sys}_{\Sigma_0}$, given by Algorithm 1, is open, then Σ_0 is satisfiable. That is, $\mathcal{A}^{sys}_{\Sigma_0}$ has at least one open bunch that allows us to construct a model of Σ_0. The final step of proving the completeness of the tableau method is establishing its termination.

7.1 Open Bunch Model Construction

In this subsection, we define a method to associate a Kripke structure to any open bunch of the systematic tableau $\mathcal{A}^{sys}_{\Sigma_0}$. Later, we prove that this Kripke structure is a model of Σ_0.

Definition 39 (Stage). *Given a branch, b of a tableaux T, a stage in T is every maximal subsequence of successive nodes $n_i, n_{i+1}, \dots, n_j$ in b such that $\tau(n_k)$ is not a $(\mathsf{Q}\circ)$-child of $\tau(n_{k-1})$, for all k such that $i < k \le j$. We denote by $\mathsf{stages}(b)$ the sequence of all stages of b. The successor relation on $\mathsf{stages}(b)$ is induced by the successor relation on b. The labelling function τ is extended to stages as the union of the original τ applied to every node in a stage.* ∎

We prove that any open bunch of the systematic tableau $\mathcal{A}^{sys}_{\Sigma_0}$ represents a model of the initial set of formulae Σ_0.

Definition 40 ($\alpha\beta^+$-saturated Stage). *A set of state formulae Ψ is $\alpha\beta^+$-saturated* iff *for all $\sigma \in \Psi$:*

1. *If σ is an α-formula then $S_\sigma \subseteq \Psi$*
2. *If σ is a β-formula of range k , but it is not a β^+-formula, then $S_{\beta_i} \subseteq \Psi$ for some $1 \le i \le k$.*
3. *If σ is a β-formula and also a β^+-formula of range k then either $S_{\beta_i} \subseteq \Psi$ or $S^+_{\Sigma,\beta_i} \subseteq \Psi$ for some $1 \le i \le k$ and $\Sigma = \tau(n) \setminus \{\sigma\}$ for some $n \in s$.*

We say that a stage s in $\mathcal{A}^{sys}_{\Sigma_0}$ is $\alpha\beta^+$-saturated iff *$\tau(s)$ is $\alpha\beta^+$-saturated. For a given set Σ of state-formulae, we denote by $\mathsf{Comp}(\Sigma)$ the union of all the minimal sets that contains Σ and are $\alpha\beta^+$-saturated.*
∎

Definition 41 (Expanded Bunch and Tableau). *An open branch b is expanded if each stage $s \in \mathsf{stages}(b)$ is $\alpha\beta^+$-saturated and b is eventuality-covered. A bunch is expanded if all its open branches are expanded. A tableau is expanded if all its open bunches are expanded.* ∎

Proposition 42. *Given any set of state formulae Σ_0, the systematic tableau $\mathcal{A}^{sys}_{\Sigma_0}$ is expanded.*

Proof. Trivial, by construction. ∎

Definition 43 (Open Bunch Model Construction). *For any expanded bunch H of $\mathcal{A}^{sys}_{\Sigma_0}$, we define the Kripke-structure $\mathcal{K}_H = (S, R, L)$ such that $S = \bigcup_{b \in H} \mathsf{stages}(b)$ and for any $s \in S$: $L(s) = \{p \mid p \in \tau(n) \cap \mathsf{Prop}$ for some node $n \in s\}$; and R is the relation induced in $\mathsf{stages}(b)$ for each $b \in H$.* ∎

7.2 Properties of the Open Branches of $\mathcal{A}^{sys}_{\Sigma_0}$

In order to prove that $\mathcal{K}_H$, as defined in Definition 43, is a model for the label of the root of H, we first prove the required auxiliary properties of the systematic construction of the tableau $\mathcal{A}^{sys}_{\Sigma_0}$. The systematic construction of Algorithm 1 produces uniform sets as expandable nodes. The selection is always made in uniform sets and loop-nodes are also labelled by uniform sets.

Remark 44 (Notation for Eventualities and Tableau Rule Application). In what follows, we use χ to represent a formula of one of the two following forms: $(\sigma_2 \wedge \Diamond\sigma_3)$ or $\Box\sigma_2$. Then, $\sigma_1 \,\mathcal{U}\, \chi$ stands for one of the two possible eventualities. We say that the *corresponding β^+ rule* is applied to a selected $\mathsf{A}\Pi$ with some marked eventuality $\pi \in \Pi$, meaning that $(\mathsf{A}\mathcal{U}\,\sigma)^+$ is applied when π is $\sigma_1 \,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$, $(\mathsf{A}\mathcal{U}\,\Box)^+$ is applied when π is $\sigma_1 \,\mathcal{U}\,\Box\sigma_2$, and $(\mathsf{A}\Box\mathcal{U}\,)$ followed by $(\mathsf{A}\mathcal{U}\,\sigma)^+$ is applied when π is $\Box(\sigma_1 \,\mathcal{U}\,\sigma_2)$. We say that the *corresponding* β^+ *rule* is applied to a selected $\mathsf{E}\Pi$ meaning that $(\mathsf{E}\mathcal{U}\,\sigma\Box)^+$ is applied when Π contains at least one $\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$ or at lest one $\sigma_1\,\mathcal{U}\,\Box\sigma_2$, and otherwise when Π contains at least one $\Box(\sigma_1\,\mathcal{U}\,\sigma_2)$, then $(\mathsf{E}\mathcal{U}\,\sigma\Box)^+$ is applied just after $(\mathsf{E}\Box\mathcal{U}\,)$ has been applied to every $\Box(\sigma_1\,\mathcal{U}\,\sigma_2)$ in Π. For clarity, we consider sets S_{Σ,β_i} and S^+_{Σ,β_i} used in the tableau rules creating child nodes from left ($i = 1$) to right, where i is the rank of the rule. ∎

Definition 45 (Variants). *For a given set Π of basic path formulae, we denote by $\mathsf{Variants}(\Pi)$ the collection of all subsets of the sets Π' that are obtained from Π by one simultaneous application of any number (including zero) of individual substitutions of some $\pi \in \Pi$ by π' that satisfies the following rules:*

- *π is an eventuality $\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$ and π' is either $\Diamond\sigma_3$ or a next-step variant of π.*
- *π is an eventuality $\sigma_1\,\mathcal{U}\,\Box\sigma_2$ and π' is either $\Box\sigma_2$ or a next-step variant of π.*
- *π is $\Box(\sigma_1 \vee \Box\sigma_2)$ and π' is $\Box\sigma_2$.* ∎

The following two propositions establish general properties of the rule-based decomposition of E-conjunctive and A-disjunctive formulae (respectively) along open branches.

Proposition 46. *Let b be an open branch of $\mathcal{A}^{sys}_{\Sigma_0}$, let s_i, s_j ($i < j$) be any pair of consecutive stages in $\mathsf{stages}(b)$ and let $\Sigma \cup \{\mathsf{E}\Pi\}$ be the uniform set labelling the first node of s_i. There exists a (possibly empty and minimal) uniform set $\Pi' \in \mathsf{Variants}(\Pi)$ such that $\mathsf{E}\Pi' \in \tau(s_j)$ and for all $\pi \in \Pi$:*

(a) if $\pi = \sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$ then there exists $k \geq i$ such that $\sigma_1 \in \tau(s_h)$ for all $h \in i..k-1$ and
- *(a1) $k < j$ and $\sigma_2 \in \tau(s_k)$ and $\sigma_3 \in \tau(s_{k'})$ for some k' such that $k \leq k' \leq j$, or*
- *(a2) $k < j$ and $\sigma_2 \in \tau(s_k)$ and $\Diamond\sigma_3 \in \Pi'$, or*
- *(a3) $k = j$ and π or some next-step variant of it is in Π'.*

(b) if $\pi = \sigma_1\,\mathcal{U}\,\Box\sigma_2$ then there exists $k \geq i$ such that $\sigma_1 \in \tau(s_h)$ for all $h \in i..k-1$ and
- *(b1) $k < j$ and $\sigma_2 \in \tau(s_h)$ for some h such that $i \leq h \leq k$ and $\Box\sigma_2 \in \Pi'$, or*
- *(b2) $k = j$ and $\sigma_1\,\mathcal{U}\,\Box\sigma_2$ or some next-step variant of it is in Π'.*

(c) if $\pi = \Box(\sigma_1\,\mathcal{U}\,\sigma_2)$ then $\Box(\sigma_1\,\mathcal{U}\,\sigma_2) \in \Pi'$ and for all $h \in i..j-1$: $\sigma_2 \in \tau(s_h)$ or $\sigma_1 \in \tau(s_h)$.

(d) if $\pi = \Box(\sigma_1 \vee \Box\sigma_2)$ then there exists $k \geq i$ such that $\sigma_1 \in \tau(s_h)$ for all $h \in i..k-1$
- *(d1) $k = j$ and $\Box(\sigma_1 \vee \Box\sigma_2) \in \Pi'$, or*
- *(d2) $k < j$ and there exists $k' \in k..j-1$ such that $\sigma_2 \in \tau(s_h)$ for all $h \in k'..j-1$ and $\Box\sigma_2 \in \Pi'$*

Proof. By simultaneous induction on Π, using the rules for the E-conjunctive formulae. Note that any stage in b is $\alpha\beta^+$-saturated and the procedure $\mathsf{Uniform\_Tableau}$ applies exactly once between the last state of s_i and the first state of s_{i+1}. More specifically, in (a) we use $(\mathsf{E}\mathcal{U}\,\sigma)$ and $(\mathsf{E}\mathcal{U}\,\sigma\Box)^+$, and (a1) and (a2) come from the second application of $(\mathsf{E}\mathcal{U}\,\sigma)$ to $\mathsf{E}(\Diamond\sigma_3, \dots)$ where $\Diamond\sigma_3$ abbreviates $\mathtt{T}\,\mathcal{U}\,\sigma_3$. Similarly, item (b) comes from rules $(\mathsf{E}\mathcal{U}\,\Box)$ and $(\mathsf{E}\mathcal{U}\,\sigma\Box)^+$; Items (c) and (d) are respectively obtained from the rules $(\mathsf{E}\Box\mathcal{U}\,)$, $(\mathsf{E}\mathcal{U}\,\sigma)$, and $(\mathsf{E}\mathcal{U}\,\sigma\Box)^+$; and $(\mathsf{E}\Box\sigma)$. It is worth noting that Π' is empty if all the formulae in Π satisfy the case (a1). ∎

Proposition 47. *Let b be an open branch of $\mathcal{A}^{sys}_{\Sigma_0}$, let s_i, s_j ($j > i$) be any pair of consecutive stages in $\mathsf{stages}(b)$ and let $\Sigma \cup \{\mathsf{A}\Pi\}$ be the uniform set labelling the first node of s_i. If there exists a non-empty uniform set $\Pi' \in \mathsf{Variants}(\Pi)$ such that $\mathsf{A}\Pi' \in \tau(s_j)$, then the following four facts hold:*

(a) For all $\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3) \in \Pi$:
- *(a1) if $\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$ or a next-step variant of it is in Π', then $\sigma_1 \in \tau(s_h)$ for all $h \in i..j-1$,*
- *(a2) if $\Diamond\sigma_3 \in \Pi'$ then $\sigma_2 \in \tau(s_k)$ for some $k \in i..j-1$ and $\sigma_1 \in \tau(s_h)$ for all $h \in i..k-1$.*

(b) For all $\sigma_1\,\mathcal{U}\,\Box\sigma_2 \in \Pi$: if $\Box\sigma_2 \in \Pi'$ or $\sigma_1\,\mathcal{U}\,\Box\sigma_2$ or a next-step variant of $\sigma_1\,\mathcal{U}\,\Box\sigma_2$ is in Π', then $\{\sigma_1, \sigma_2\} \subseteq \tau(s_h)$ for all $h \in i..j-1$.

(c) For all $\Box(\sigma_1\,\mathcal{U}\,\sigma_2) \in \Pi \cap \Pi'$: $\sigma_1 \in \tau(s_h)$ or $\sigma_2 \in \tau(s_h)$ for all $h \in i..j-1$.

(d) For all $\Box(\sigma_1 \vee \Box\sigma_2) \in \Pi$,
- *(d1) if $\Box(\sigma_1 \vee \Box\sigma_2) \in \Pi$ then $\sigma_1 \in \tau(s_h)$ for all $h \in i..j-1$.*
- *(d2) if $\Box\sigma_2 \in \Pi'$ then for some $k \in i..j-1$ $\{\neg\sigma_1, \sigma_2\} \subseteq \tau(s_k)$ and $\sigma_1 \in \tau(s_h)$ for all $h \in i..k-1$ and $\sigma_2 \in \tau(s_h)$ for all $h \in k..j-1$.*

Moreover, if there exists no $\mathsf{A}\Pi' \in \tau(s_j)$ such that $\Pi' \in \mathsf{Variants}(\Pi)$, then there exists some $\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3) \in \Pi$ and some $k, k' \in i..j-1$ such that $k \leq k'$ and $\sigma_2 \in \tau(s_k)$ and $\sigma_3 \in \tau(s_{k'})$.

Proof. By simultaneous induction on Π. For the proof we note that any stage in b is $\alpha\beta^+$-saturated, and the following rules for A-disjunctive formulae hold: $(\mathsf{A}\,\mathcal{U}\,\sigma)$ and $(\mathsf{A}\,\mathcal{U}\,\sigma)^+$in (a); $(\mathsf{A}\,\mathcal{U}\,\Box)$ and $(\mathsf{A}\,\mathcal{U}\,\Box)^+$ in (b); $(\mathsf{A}\Box\,\mathcal{U}\,)$, $(\mathsf{A}\,\mathcal{U}\,\sigma)$, and $(\mathsf{A}\,\mathcal{U}\,\sigma)^+$ in (c); and $(\mathsf{A}\Box\sigma)$ in (d). It is worth noting that the rules $(\mathsf{A}\,\mathcal{U}\,\sigma)$ and $(\mathsf{A}\Box\sigma)$ generate a child where the A-disjunctive formula that comes from $\mathsf{A}\Pi$ is of the form $\mathsf{A}\Pi'$ with $\Pi' \subset \Pi$ (one formula is lost). ∎

From the previous two propositions it is easy to conclude that, for a given initial node (of stage s_i) labelled by a uniform set $\Sigma \cup \{\mathsf{Q}\Pi\}$, the mentioned sets Π' such that $\mathsf{Q}\Pi' \in \tau(s_{i+1})$ exclusively consist of path subformulae and next-step variants of the formulas in Π. It also follows that $\tau(s_{i+1})$ exclusively contains Boolean combinations of state subformulae in $\Sigma \cup \{\mathsf{Q}\Pi\}$ and the formula $\mathsf{Q}\Pi'$. The next proposition shows that the selection of an E-conjunctive formula in any open branch of $\mathcal{A}^{sys}_{\Sigma_0}$ can only be kept along finitely many stages.

Proposition 48. *Let b be an open branch of $\mathcal{A}^{sys}_{\Sigma_0}$, let $s_i \in \mathsf{stages}(b)$ and let $\Sigma \cup \{\mathsf{E}\Pi\}$ be the uniform set labelling the first node of s_i where the selectable formula $\mathsf{E}\Pi$ is selected. If $\mathsf{E}\Pi$ has the highest priority, then*

(a) For all $\pi = \sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3) \in \Pi$: $\{\sigma_2, \mathsf{E}(\Diamond\sigma_3, \Pi')\} \subseteq \tau(s_k)$ for some stage $s_k \in \mathsf{stages}(b)$ ($k \geq i$) and some $\Pi' \in \mathsf{Variants}(\Pi \setminus \{\pi\})$.
(b) For all $\pi = \sigma_1\,\mathcal{U}\,\Box\sigma_2 \in \Pi$: $\mathsf{E}(\Box\sigma_2, \Pi') \in \tau(s_k)$ for some stage $s_k \in \mathsf{stages}(b)$ ($k \geq i$) and some $\Pi' \in \mathsf{Variants}(\Pi \setminus \{\pi\})$.

and if $\mathsf{E}\Pi$ has the lowest priority, then

(c) For all $\pi = \Box(\sigma_1\,\mathcal{U}\,\sigma_2) \in \Pi$: $\sigma_2 \in \tau(s_k)$ for some stage $s_k \in \mathsf{stages}(b)$ ($k \geq i$).

Proof. Suppose that $\mathsf{E}\Pi$ has the highest priority, then Π contains at least one $\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$ or $\sigma_1\,\mathcal{U}\,\Box\sigma_2$, then the β^+-rule $(\mathsf{E}\,\mathcal{U}\,\sigma\Box)^+$ is applied in the first node of stage s_i where $\mathsf{E}\Pi$ is selected and the resulting $\mathsf{E}\Pi'$ of the highest priority in its children are kept selected, hence $(\mathsf{E}\,\mathcal{U}\,\sigma\Box)^+$ is again applied to them, and so on. We proceed by contradiction, supposing that (a) and (b) does not hold. We get a contradiction from the hypothesis that (a) does not hold, i.e. for some $\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3) \in \Pi$: $\{\sigma_2, \mathsf{E}(\Diamond\sigma_3, \Pi')\} \not\subseteq \tau(s_k)$ for every stage $s_k \in \mathsf{stages}(b)$ such that $k \geq i$ and all $\Pi' \in \mathsf{Variants}(\Pi \setminus \{\pi_i\})$. The proof for the case where (b) does not hold is identical. The fact that (a) does not hold means that at the first node of every stage $s_j \in \mathsf{stages}(b)$ such that $j \geq i$ there is a selected formula $\mathsf{E}\Pi_{s_j}$ which satisfies that $\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$ or some step-variant of it is in Π_{s_j}. Hence, except for a finite number of applications of $(\mathsf{E}\,\mathcal{U}\,\sigma\Box)^+$ that extends the branch b with some set $S^+_{\Sigma_{s_j},\beta_i}$ (where Σ_{s_j} is the context of the selected formula at the first node of each stage s_j, in particular $\Sigma_{s_i} = \Sigma$) such that $1 \leq i \leq n$, the branch b is repeatedly extended with some set $S^+_{\Sigma_{s_j},\beta_i}$ such that $n+1 \leq i \leq 2n$, which includes a next-step variant of at least one formula in Π_{s_j}. Note that this next-step variant could be of $\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$ or some other formula of the form $\sigma\,\mathcal{U}\,(\sigma \wedge \Diamond\sigma)$ or $\sigma\,\mathcal{U}\,\Box\sigma$. In any case, the uniform set labelling the first node of each stage s_j ($i \leq j$) has the form $\Sigma_{s_j}, \mathsf{E}\Pi_{s_j}$ where Π_{s_j} contains at least one next-step variant of $\sigma_1\,\mathcal{U}\,(\sigma_2 \wedge \Diamond\sigma_3)$, $\mathsf{E}\Pi_{s_j}$ is the selected formula and for every (except a finite number of) Σ_{s_k} where $i \leq k \leq j$, Π_{s_j} contains a formula of the form $\sigma \wedge (\neg\Phi_1 \wedge \cdots \wedge \neg\Phi_r))\,\mathcal{U}\,\chi$ such that $\Phi_h = \Sigma_{s_k}$ for some $1 \leq h \leq r$. Since no other β^+-rule is applied each Σ_{s_j} is a subset of the finite set formed by all state formulae that are subformulae of some formula in $\Sigma_{s_i} \cup \Pi$ and their negations. Hence, there exists a finite number of different Σ_{s_j}. Therefore, after finitely many applications of the β^+-rule $(\mathsf{E}\,\mathcal{U}\,\sigma\Box)^+$, for some $k >= i$, $\Sigma_{s_k} = \Sigma_{s_h}$ for some $h \in \{i, ..k-1\}$, and $\neg\Sigma_{s_k} \in \tau(s_k)$, hence, Σ_{s_k} must be inconsistent. Since b is open, this is a contradiction. It means that for some $k \geq i$, the application of the corresponding β^+-rule $(\mathsf{E}\,\mathcal{U}\,\sigma\Box)^+$ should produce a node whose label contains $S^+_{\Sigma_{s_j},\beta_i}$ where $1 \leq i \leq n$. Henceforth, the open branch b must satisfy (a).
If $\mathsf{E}\Pi$ has the lowest priority, then Π contains at least one $\Box(\sigma\,\mathcal{U}\,\sigma)$ (and none $\sigma\,\mathcal{U}\,(\sigma \wedge \Diamond\sigma)$ and none $\sigma\,\mathcal{U}\,\Box\sigma$). Let us suppose there are $n \geq 1$ formulae, i.e. $\mathsf{E}\Pi = \mathsf{E}(\Box(\sigma_1^1\,\mathcal{U}\,\sigma_2^1), \ldots, \Box(\sigma_1^n\,\mathcal{U}\,\sigma_2^n), \Pi')$ for some Π' that does not contain any eventuality. Then, the α-rule $(\mathsf{E}\Box\,\mathcal{U}\,)$ is applied n times transforming the selected $\mathsf{E}\Pi$ into

$$\mathsf{E}(\sigma_1^1\,\mathcal{U}\,\sigma_2^1, \ldots, \sigma_1^n\,\mathcal{U}\,\sigma_2^n, \circ\Box(\sigma_1^1\,\mathcal{U}\,\sigma_2^1), \ldots, \circ\Box(\sigma_1^n\,\mathcal{U}\,\sigma_2^n), \Pi')$$

which has the highest priority. Hence, by a particular application of the case (a), for all $1 \le j \le n$: $\sigma_2^j \in \tau(s_k)$ for some stage $s_k \in \mathsf{stages}(b)$ $(k \ge i)$. ∎

For A-disjunctive formulae, not only the outer context, but also the inner context, plays an important role. The next propositions explain the role of both kinds of contexts.

Proposition 49. *Let b be an open branch of $\mathcal{A}^{sys}_{\Sigma_0}$, let $s_i \in \mathsf{stages}(b)$ and let $\Sigma_{s_i} \cup \{\mathsf{A}\Pi\}$ be the uniform set labelling the first node of s_i where $\mathsf{A}\Pi$ is selected and some eventuality $\pi_{\mathcal{U}} \in \Pi$ is marked. Let b be any branch where the next-steps variants of $\pi_{\mathcal{U}}$ are successively marked and $\Pi_{s_i} = \Pi \setminus \{\pi_{\mathcal{U}}\}$ is the initial inner context. Then, one of the following two facts hold:*

(a) There exists $k \ge i$ and some $\Pi' \in \mathsf{Variants}(\Pi \setminus \{\pi_{\mathcal{U}}\})$ such that:
- *(a1) $\{\sigma_2, \mathsf{A}(\Diamond\sigma_3, \Pi')\} \subseteq \tau(s_k)$ if $\pi_{\mathcal{U}} = \sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3)$.*
- *(a2) $\mathsf{A}(\Box\sigma_2, \Pi') \in \tau(s_k)$ if $\pi_{\mathcal{U}} = \sigma_1 \mathcal{U} \Box\sigma_2 \in \Pi'$.*
- *(a3) $\sigma_2 \in \tau(s_k)$ if $\pi_{\mathcal{U}} = \Box(\sigma_1 \mathcal{U} \sigma_2) \in \Pi'$.*

(b) There exists $k \ge i$ such that the first node of s_k is a loop-node whose companion node is in s_h, for some $h \in \{i..k-1\}$, some $\Pi' \in \mathsf{Variants}(\Pi \setminus \{\pi_{\mathcal{U}}\})$ and some next-step variant $\pi^v_{\mathcal{U}}$ of $\pi_{\mathcal{U}}$ such that $\mathsf{A}(\pi^v_{\mathcal{U}}, \Pi') \in \tau(s_j)$ for all $j \in \{h..k\}$, and $\varphi_{\Pi'} \in \tau(s_k)$

Moreover, in both cases,for all $j \in \{i..k-1\}$:

1. *$\sigma_1 \in \tau(s_j)$ for all $\sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3) \in \Pi' \cup \{\pi_{\mathcal{U}}\}$,*
2. *$\{\sigma_1, \sigma_2\} \subseteq \tau(s_j)$ for all $\sigma_1 \mathcal{U} \Box\sigma_2 \in \Pi' \cup \{\pi_{\mathcal{U}}\}$,*
3. *$\sigma_1 \in \tau(s_j)$ or $\sigma_2 \in \tau(s_j)$ for all $\Box(\sigma_1 \mathcal{U} \sigma_2) \in \Pi' \cup \{\pi_{\mathcal{U}}\}$, and*
4. *$\sigma_1 \in \tau(s_j)$ for all $\Box(\sigma_1 \vee \Box\sigma_2) \in \Pi'$.*

Proof. If $\varphi_{\pi_{s_i}} = \mathbf{F}$, the uniform set labelling the first node at each stage s_j $(j \ge i)$ of b has the form

$$\Sigma_{s_j}, \mathsf{A}((\sigma_1 \wedge (\neg\Sigma_{s_i} \wedge \cdots \wedge \neg\Sigma_{s_{j-1}}))\,\mathcal{U}\,\chi, \Pi_{s_j})$$

where each Σ_{s_j} is the outer (resp. inner) context of the selected formula containing the marked next-step variant of $\pi_{\mathcal{U}}$) at the first node of each stage s_j. In particular, $\Sigma_{s_i} = \Sigma$. Since no other β^+-rule is applied each Σ_{s_j} is a subset of the finite set formed by all state formulae that are subformulae of some formula in $\Sigma_{s_i} \cup \Pi$ and their negations. Hence, there are a finite number of different Σ_{s_j}. Therefore, after finitely many applications of the β^+-rule, $\Sigma_{s_h} = \Sigma_{s_j}$, for some $h >= i$, for some $j \in \{i,..h-1\}$, and $\sigma_1 \wedge (\neg\Sigma_{s_i} \wedge \cdots \wedge \neg\Sigma_{s_{h-1}}) \in \tau(s_h)$. In particular, $\neg\Sigma_{s_h} \in \tau(s_h)$, hence, Σ_{s_h} must be inconsistent. Since b is open, this is a contradiction. This means that, for some $k \ge i$ the application of the corresponding β^+-rule should produce a node generated by the corresponding set $S^+_{\beta_1}$. Henceforth, the open branch b must satisfy (a1) or (a2) or (a3), depending on the case of $\pi_{\mathcal{U}}$.

If $\varphi_{\pi_{s_i}} \ne \mathbf{F}$, then, the uniform set labelling the first node at each stage s_j $(j \ge i)$ has the form

$$\Sigma_{s_j}, \mathsf{A}((\sigma_1 \wedge (\neg\Sigma_{s_i} \vee \varphi_{\Pi_{s_i}}) \wedge \cdots \wedge (\neg\Sigma_{s_{j-1}} \vee \varphi_{\Pi_{s_{j-1}}}))\,\mathcal{U}\,\chi, \Pi_{s_j})$$

where each Π_{s_h} is the inner context at the first node of each stage s_h. In particular, $\Pi_{s_i} = \Pi \setminus \{\pi_{\mathcal{U}}\}$. Since no other β^+-rule is applied, then

- each Σ_{s_j} is a subset of the following finite set $LC(\Sigma, \Pi)$: $LC(\Sigma, \Pi)$ is formed by all state formulae that are subformulae of some formula in $\Sigma \cup \Pi$, their negations, and a formula $\mathsf{E}\Diamond\sigma_2$ for each subformula $\Box(\sigma_1 \mathcal{U} \sigma_2)$ in Π (see Definition 19), and
- each Π_{s_j} is a subset of the finite set of all state formulae that are subformulae of some formula in Π. Indeed, each $\Pi_{s_{j+1}} \in \mathsf{Variants}(\Pi_{s_j})$ for all $j \ge i$.

In particular, there are a finite number of different outer and inner contexts. Henceforth, there are two possibilities. First, for some h, k such that $k > h \ge i$, both $\Sigma_{s_k} = \Sigma_{s_h}$ and $\Pi_{s_k} = \Pi_{s_h}$; and second for some $h \ge i$, the formula $\varphi_{\pi_{s_h}}$ is $\mathbf{F}$. In the latter case, the item (a) must be satisfied for some $k \ge h$. In the former case, by Definition 45 and Proposition 47, for all $j \in \{h..k\}$: $\Pi_{s_j} = \Pi_{s_k}$. Let $\Pi' = \Pi_{s_k}$, then the first nodes at the sequence of stages $s_h, s_{h+1}, \ldots, s_k$ are respectively labelled by

$$\Sigma_{s_h} \cup \{\mathsf{A}((\sigma_1 \wedge \delta)\,\mathcal{U}\,\chi, \Pi')\}, \Sigma_{s_{h+1}} \cup \{\mathsf{A}((\sigma_1 \wedge \delta)\,\mathcal{U}\,\chi, \Pi')\}, \ldots, \Sigma_{s_k} \cup \{\mathsf{A}((\sigma_1 \wedge \delta)\,\mathcal{U}\,\chi, \Pi')\}$$

where $\delta = (\neg\Sigma_{s_i} \vee \varphi_{\Pi_{s_i}}) \wedge \cdots \wedge (\neg\Sigma_{s_h} \vee \varphi_{\Pi'}) \wedge \cdots \wedge (\neg\Sigma_{s_{k-1}} \vee \varphi_{\Pi'})$ or equivalently $\delta = (\neg\Sigma_{s_i} \vee \varphi_{\Pi_{s_i}}) \wedge \cdots \wedge ((\neg\Sigma_{s_h} \wedge \cdots \wedge \neg\Sigma_{s_{z-1}}) \vee \varphi_{\Pi'})$. Hence, in node s_k, the application of the β^+-rule to the marked eventuality produces a right-hand child that contains Σ_{s_k} and $\sigma_1 \wedge \delta$. Therefore, by rules $(\wedge)$ and $(\vee)$, it also contains $\neg\Sigma_{s_h} = \neg\Sigma_{s_k}$. Therefore, since b is open, $\tau(s_k)$ must contain $\varphi_{\Pi'}$, which completes the proof of item (b). Moreover, in both cases (a) and (b), for all $j \in \{i..k-1\}$, each inner context Π_{s_j+1} satisfies the properties of Π' in Proposition 46 with respect to Π_{s_j} as Π. Consequently, the last four items of the proposition hold. ∎

The next two propositions provide a detailed description of how the highest priority formulae evolve in open branches.

Proposition 50. *Let b be an open branch of $\mathcal{A}^{sys}_{\Sigma_0}$, and let $\mathsf{E}\Pi$ be of the highest priority that is selected at some stage $s_i \in \mathsf{stages}(b)$. Then there exists a state $s_k \in \mathsf{stages}(b)$ (for some $k \geq i$) and some (possibly empty and minimal) set $\Pi' \in \mathsf{Variants}(\Pi)$ such that $\mathsf{E}\Pi' \in \tau(s_k)$, $\mathsf{E}\Pi'$ is of the lowest priority and for all $\pi \in \Pi$ the following facts hold:*

(a) If $\pi = \sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3)$ then there exists j, j' such that $i \leq j \leq j' \leq k$, for all $h \in \{i, \ldots, j-1\}$: $\sigma_1 \in \tau(s_h)$, $\sigma_2 \in \tau(s_j)$, and $\sigma_3 \in \tau(s_{j'})$.
(b) If $\pi = \sigma_1 \mathcal{U} \Box\sigma_2$ then there exists j such that $i \leq j \leq k$ and for all $h \in \{i, \ldots, j-1\}$: $\sigma_1 \in \tau(s_h)$, and for all $h \in \{j, \ldots, k\}$: $\sigma_2 \in \tau(s_h)$ and $\Box\sigma_2 \in \Pi'$.
(c) if $\pi = \Box(\sigma_1 \mathcal{U} \sigma_2)$ then $\Box(\sigma_1 \mathcal{U} \sigma_2) \in \Pi'$, and for all $j \in \{i, \ldots, k\}$: either $\sigma_1 \in \tau(s_j)$ or $\sigma_2 \in \tau(s_j)$.
(d) if $\pi = \Box(\sigma_1 \vee \Box\sigma_2)$ then one of the following two facts holds:
  (d1) For all $j \in \{i, \ldots, k\}$: $\sigma_1 \in \tau(s_j)$ and $\Box(\sigma_1 \vee \Box\sigma_2) \in \Pi'$.
  (d2) There exists j such that $i \leq j \leq k$ and for all $h \in \{i, \ldots, j-1\}$: $\sigma_1 \in \tau(s_h)$, and for all $h \in \{j, \ldots, k\}$: $\sigma_2 \in \tau(s_h)$ and $\Box\sigma_2 \in \Pi'$.

Proof. By simultaneous induction on the structures of formulae in Π and Propositions 48 and 46((a) and (b)), the above items (a) and (b) hold for some $k \geq i$. Note that in case (a), $\mathsf{E}(\Diamond\sigma_3, \Pi')$ (for some Π' such that $\{\Diamond\sigma_3\} \cup \Pi' \in \mathsf{Variants}(\Pi)$) is kept selected at stage s_j. Hence, the eventuality $\sigma_3 \in \tau(s_{j'})$ for some $j' \geq j$. Therefore, by Proposition 48, the existence of such j' is ensured. In case (b) $\mathsf{E}(\Box\sigma_2, \Pi') \in \tau(s_j)$ for some $j \geq i$, hence, by Proposition 46(d), for all $h \in \{j, \ldots, k\}$: $\sigma_2 \in \tau(s_h)$ and $\Box\sigma_2 \in \Pi'$. Items (c) and (d) are ensured by Propositions 46 (c) and (d). Additionally, by minimality, Π' only contains formulae of the forms $\Box(\sigma_1 \mathcal{U} \sigma_2)$ and $\Box(\sigma_1 \vee \Box\sigma_2)$, hence $\mathsf{E}\Pi'$ is of the priority. ∎

The other kind of the highest-priority formulae are $\mathsf{A}\Pi$ such that Π is exclusively formed by formulae of the form $\sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3)$.

Proposition 51. *Let b be an open branch of $\mathcal{A}^{sys}_{\Sigma_0}$, and let $\mathsf{A}\Pi$ be a formula of the highest priority that is selected at some stage $s_i \in \mathsf{stages}(b)$. Then there exists $\pi = \sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3) \in \Pi$ and some stage $s_k \in \mathsf{stages}(b)$ (for some $k \geq i$) such that for all $j \in \{i, \ldots, k-1\}$: $\sigma_1 \in \tau(s_j)$, $\{\sigma_2, \mathsf{A}(\Diamond\sigma_3, \Pi')\} \subseteq \tau(s_k)$ for some $\Pi' \in \mathsf{Variants}(\Pi \setminus \{\pi\})$. Moreover, $\mathsf{A}(\Diamond\sigma_3, \Pi')$ is also a formula of the highest priority.*

Proof. According to Definitions 24 and 26, one eventuality in Π must be marked at the stage s_i of b. Hence, there exists $\sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3) \in \Pi$ that is the marked eventuality at stage s_i. Since φ_Π is $\mathbf{F}$ when Π is exclusively formed by formulae of the form $\sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3)$, the item (a) of Proposition 49 holds. Hence, by Proposition 49 (a1), there exists $k \geq i$ and Π' such that $\{\sigma_2, \mathsf{A}(\Diamond\sigma_3, \Pi')\} \in \tau(s_k)$. Since $\Pi' \in \mathsf{Variants}(\Pi \setminus \{\pi\})$, every formula in Π' is of the form $\sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3)$ (in particular, $\Diamond\sigma$ which abbreviates $\mathbf{T} \mathcal{U} (\sigma \wedge \Diamond\mathbf{T})$). Hence, $\mathsf{A}(\Diamond\sigma_3, \Pi')$ is also of the highest-priority. ∎

Next, we show that any open branch of $\mathcal{A}^{sys}_{\Sigma_0}$ is eventuality-covered. In the sequel, we deal with uniform sets formed by non-selectable and the lowest priority formulae (i.e, without the highest-priority formulae), we call them *cycle-uniform* sets.

Proposition 52. *Any open branch b of $\mathcal{A}^{sys}_{\Sigma_0}$ is eventuality-covered.*

Proof. Let b be any open branch of $\mathcal{A}^{sys}_{\Sigma_0}$, we are going to show that there must exist some stage s_ℓ in b with the first node n_ℓ labelled by a cycle-uniform set Σ_ℓ such that any selection of a formula of the lowest priority in Σ_ℓ produces a loop-node whose companion node is n_ℓ.

Let Σ and Σ' be any two cycle-uniform sets of state formulae, we say that $\Sigma' \preceq \Sigma$ iff every formula in Σ' is either a proper subformula of some formula $\mathsf{Q}\Pi \in \Sigma$ or its negation, or a formula $\mathsf{Q}\Pi'$ such that there exists $\mathsf{Q}\Pi \in \Sigma$ such that $\Pi' \in \mathsf{Variants}(\Pi)$. It is worth noting that the formulae $\mathsf{E}\Diamond\sigma_2$ that can be introduced by φ_Π (see Definition 19) are of the highest priority, then they cannot belong to any cycle-uniform set. Let $b = s_0, \dots, s_i, \dots, s_j, \dots$ $(0 \geq i > j)$ be any open branch of $\mathcal{A}^{sys}_{\Sigma_0}$, and let Σ and Σ' respectively be the cycle-uniform sets labelling the first node of s_i and s_j. By Propositions 46, 47, 48 and 49, $\Sigma' \preceq \Sigma$. Moreover, for any cycle-uniform set Σ, $\preceq$ is a well-founded order on the collection of all cycle-uniform sets Σ' such that $\Sigma' \preceq \Sigma$.

Let $b = s_0, \dots, s_i, \dots, s_j, \dots, s_k, \dots$ $(0 \geq i > j \geq k)$ be any open branch of $\mathcal{A}^{sys}_{\Sigma_0}$. Suppose that every highest priority formula in the initial uniform set has been selected before the stage s_i. Let Σ be the cycle-uniform set labelling the first node of s_j which is a loop-node whose companion is the first node of s_i. Suppose that Σ contains at least one lowest priority formula $\mathsf{A}\Pi$ that was selected at s_i. If b is not already eventuality-covered, this means that there exists $\mathsf{A}\Pi' \in \Sigma$ of the lowest priority that has not been selected. Suppose that $\mathsf{A}\Pi'$ is selected at s_j and there exists a loop-node at s_k labelled by Σ' then, $\Sigma' \preceq \Sigma$. If $\Sigma = \Sigma'$ and there are no more selectable formula in Σ, b is already eventuality covered and the first node of s_i is n_ℓ. Otherwise, $\Sigma' \prec \Sigma$, so that the companion node of the first node of s_k is the first node of some stage s_h such that $h > i$. In general, for any number of the lowest priority formulae in Σ, by well-foundness of $\prec$ there should exist a minimal node n_ℓ labelled by a cycle-uniform set Σ_ℓ such that any selection of the lowest priority formula in Σ_ℓ produces a loop-node whose companion node is n_ℓ. Hence, the branch b ends by a subsequence of $n \geq 2$ (possibly non-consecutive) stages $s_{i_1}, \dots, s_{i_n}$ whose first node is labelled by Σ_ℓ, where n_ℓ is the first node of s_{i_1}, and each selectable (lowest priority) formula in Σ_ℓ is selected at some stage in $s_{i_1}, \dots, s_{i_n}$. In particular, Σ_ℓ could be empty, then $n = 2$ and b is trivially eventuality-covered. ∎

7.3 Refutational Completeness

In this subsection we prove that our tableau method is refutationally complete, that is if a set of state formulae is unsatisfiable then there exists a closed tableau for it. For that, we first ensure the existence of a model for any open bunch of $\mathcal{A}^{sys}_{\Sigma_0}$.

Lemma 53 (Model Existence). *Let H be an expanded bunch of $\mathcal{A}^{sys}_{\Sigma_0}$ and $\mathcal{K}_H = (S, R, L)$ be as in Definition 43. For every state $s \in S$, if $\sigma \in L(s)$ then $\mathcal{K}_H, s, 0 \models \sigma$. Therefore, $\mathcal{K}_H \models \Sigma$.*

Proof. Let H be any expanded bunch of $\mathcal{A}^{sys}_{\Sigma_0}$ and let b be any open branch in H. The construction of any branch of $\mathcal{A}^{sys}_{\Sigma_0}$ starts by selecting a formula of the highest-priority (if any) and marking eventualities as explained in Definition 24. At most one eventuality is marked inside the unique selected $\mathsf{Q}\Pi$ and the rules of Figure 7 are applied to this formula and to no one else. When a β^+-rule is applied to a formula of the highest priority (independently of the marked eventuality), then only the outer context (but no the inner context) is used to construct the new-step variant. Therefore, while some highest priority formulae is selected, previous labels cannot be repeated. Consequently, the initial segment of any open branch has no loop-nodes. This initial segment can be empty or not. According to Proposition 52, the branch b is eventuality covered. Hence, there exists a (possibly empty) cycle-uniform set Σ_ℓ such that for some $i \geq 0$: $b = s_0, s_1, \dots, s_{i-1}, s_i, s_{i+1}, \dots, s_j, n_\ell$, where each s_h stands for a stage and n_ℓ is a non-expandable loop-node labelled by Σ_ℓ whose companion node is the first node at stage s_i. Let $s_{i_1}, s_{i_2}, \dots, s_{i_z}$ be a the subsequence formed by all the stages in $s_i, s_{i+1}, \dots, s_j$ whose first node is labelled by Σ_ℓ (in particular, $s_{i_1} = s_i$). Then, each lowest priority formula in Σ_ℓ has been selected at some node n_h $(h \in i..j + 1)$. The tableau branch b represents a cyclic branch (of a model) such that $\mathsf{path}(b) = s_0, s_1, \dots, s_{i-1}\langle s_i, s_{i+1}, \dots, s_j\rangle^\omega$ where each state s_h $(h \in 0..j)$ is labelled by the set of all atoms occurring in the label $\tau(s_h)$ of the tableau stage s_h (in the label of some tableau node at stage s_h).

We are going to prove that $\mathcal{K}_H, s_a, 0 \models \sigma$ for any $a \in 0...j$ and any formula σ in $\tau(s_a)$, by structural induction on the formula σ. The base of the induction $\sigma = p \in \mathsf{Prop}$ is ensured by Definition 43: $\mathcal{K}_H, s_a, 0 \models p$.

The bunch H allows us to ensure that whenever a tableau node in stage s_a is labelled by an elementary set $\{\Sigma, \mathsf{A}\circ\Phi_1, \dots, \mathsf{A}\circ\Phi_n, \mathsf{E}\circ\Psi_1, \dots, \mathsf{E}\circ\Psi_m\} \subseteq L(s)$ then, by rule $(\mathsf{Q}\circ)$, the bunch H contains one successor stage s^i_{a+1}, for each $i \in 1..m$, that contains $\{A\Phi_1, \dots, \mathsf{A}\Phi_n, \mathsf{E}\Psi_i\}$. Since by induction hypothesis, we can assume that $\mathcal{K}_H, s^i_{a+1}, 0 \models A\Phi_1, \dots, \mathsf{A}\Phi_n, \mathsf{E}\Psi_i$ for all $i \in 1..m$, and Σ is a consistent set of literals, then we can infer that $\mathcal{K}_H, s_a, 0 \models \{\Sigma, \mathsf{A}\circ\Phi_1, \dots, \mathsf{A}\circ\Phi_n, \mathsf{E}\circ\Psi_1, \dots, \mathsf{E}\circ\Psi_m\}$.

To complete the proof, we prove different cases for σ being a formula of the form $\mathsf{Q}\Pi$, depending on whether σ is selectable or not and, in the selectable case, depending on the σ priority for the selection strategy.

Let $\sigma = \mathsf{E}\Pi \in \tau(s_a)$ be non-selectable. Then every $\pi \in \Pi$ is of the form $\Box(\sigma_1 \vee \Box\sigma_2)$. Hence, by Proposition 46 (d) and the induction hypothesis, there exists a state s_k (for some $k \geq a$) and some non-empty set $\Pi' \in \mathsf{Variants}(\Pi)$ such that $\mathcal{K}_H, s_k, 0 \models \mathsf{E}\Pi'$, $\Box(\sigma_1 \vee \Box\sigma_2) \in \Pi'$, and for all $\pi \in \Pi$:

- $\mathcal{K}_H, s_j, 0 \models \sigma_1$ for all $a \leq j \leq k$, and
- there exists j such that $a \leq j \leq k$ and for all $h \in a..j-1$: $\mathcal{K}_H, s_h, 0 \models \sigma_1$, and for all $h \in j..k$: $\mathcal{K}_H, s_h, 0 \models \sigma_2$ and $\Box\sigma_2 \in \Pi'$.

Therefore, $\mathcal{K}_H, s_a, 0 \models \mathsf{E}\Pi$.

Let $\sigma = \mathsf{A}\Pi \in \tau(s_a)$ be non-selectable. Then every $\pi \in \Pi$ is of the form $\Box(\sigma_1 \vee \Box\sigma_2)$. By Proposition 47 (d) and the induction hypothesis, there exists a state s_k (for some $k \geq a$) and some non-empty set $\Pi' \in \mathsf{Variants}(\Pi)$ such that $\mathcal{K}_H, s_k, 0 \models \mathsf{A}\Pi'$ and for all $\pi = \Box(\sigma_1 \vee \Box\sigma_2) \in \Pi$:

- If $\Box(\sigma_1 \vee \Box\sigma_2) \in \Pi'$ then $\mathcal{K}_H, s_j, 0 \models \sigma_1$ for all $j \in a..k$, and
- if $\Box\sigma_2 \in \Pi'$ then there exists j such that $a \leq j \leq k$ and for all $h \in a..j-1$: $\mathcal{K}_H, s_h, 0 \models \sigma_1$, and for all $h \in j..k$: $\mathcal{K}_H, s_h, 0 \models \sigma_2$.

Therefore, $\mathcal{K}_H, s_a, 0 \models \mathsf{A}\Pi$.

Let $\sigma = \mathsf{E}\Pi \in \tau(s_a)$ be a (selectable) formula of the highest priority. According to Proposition 50 and the induction hypothesis, there exists a state s_k (for some $k \geq a$) and some (possibly empty and minimal) set $\Pi' \in \mathsf{Variants}(\Pi)$ such that $\mathcal{K}_H, s_k, 0 \models \mathsf{E}\Pi'$ and for all $\pi \in \Pi$ the following facts hold:

- If $\pi = \sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3)$ then there exists j, j' such that $a \leq j \leq j' \leq k$, for all $h \in i..j-1$: $\mathcal{K}_H, s_h, 0 \models \sigma_1$, $\mathcal{K}_H, s_j, 0 \models \sigma_2$, and $\mathcal{K}_H, s_{j'}, 0 \models \sigma_3$.
- If $\pi = \sigma_1 \mathcal{U} \Box\sigma_2$ then there exists j such that $a \leq j \leq k$ and for all $h \in a..j-1$: $\mathcal{K}_H, s_h, 0 \models \sigma_1$, and for all $h \in j..k$: $\mathcal{K}_H, s_h, 0 \models \sigma_2$ and $\Box\sigma_2 \in \Pi'$.
- If $\pi = \Box(\sigma_1 \mathcal{U} \sigma_2)$ then $\Box(\sigma_1 \mathcal{U} \sigma_2) \in \Pi'$, and for all $j \in a..k$: either $\mathcal{K}_H, s_j, 0 \models \sigma_1$ or $\mathcal{K}_H, s_a, 0 \models \sigma_2$.
- If $\pi = \Box(\sigma_1 \vee \Box\sigma_2)$ then one of the following two facts holds:
  - For all $j \in a..k$: $\mathcal{K}_H, s_j, 0 \models \sigma_1$ and $\Box(\sigma_1 \vee \Box\sigma_2) \in \Pi'$.
  - There exists j such that $a \leq j \leq k$ and for all $h \in a..j-1$: $\mathcal{K}_H, s_h, 0 \models \sigma_1$, and for all $h \in j.., k$: $\mathcal{K}_H, s_h, 0 \models \sigma_2$ and $\Box\sigma_2 \in \Pi'$.

Therefore, $\mathcal{K}_H, s_a, 0 \models \mathsf{E}\Pi$.

Let $\sigma = \mathsf{A}\Pi \in \tau(s_a)$ be a formula of the highest-priority where $\sigma_1 \mathcal{U} (\sigma_2 \wedge \Diamond\sigma_3) \in \Pi$ is marked. By Proposition 51 and the induction hypothesis, we have that $\mathcal{K}_H, s_v, 0 \models \sigma_2$ for some $v \geq a$, $\mathcal{K}_H, s_z, 0 \models \sigma_1$ for all $z \in a..v-1$. In addition, $\mathsf{A}(\Diamond\sigma_3, \Pi') \in \tau(s_v)$ (which is also the highest priority formula). Hence, by induction hypothesis, $\mathcal{K}_H, s_v, 0 \models \mathsf{A}(\Diamond\sigma_3, \Pi')$. Additionally, by Proposition 47 and the induction hypothesis:

(a) For all $\sigma'_1 \mathcal{U} (\sigma'_2 \wedge \Diamond\sigma'_3) \in \Pi$:
  - If $\sigma'_1 \mathcal{U} (\sigma'_2 \wedge \Diamond\sigma'_3)$ is also in Π', then $\mathcal{K}_H, s_z, 0 \models \sigma'_1$ for all $z \in a..v-1$.
  - If $\Diamond\sigma'_3 \in \Pi'$ then $\mathcal{K}_H, s_z, 0 \models \sigma'_2$ for some $z \in a..v-1$.

(b) For all $\sigma'_1 \mathcal{U} \Box\sigma'_2 \in \Pi$:
  - If $\sigma'_1 \mathcal{U} \Box\sigma'_2$ 'is also in Π', then $\mathcal{K}_H, s_z, 0 \models \sigma'_1$ for all $z \in a..v-1$
  - if $\Box\sigma'_2 \in \Pi'$ then there exists $j \in a..v-1$ such that $\mathcal{K}_H, s_j, 0 \models \sigma'_1$ for all $z \in a..j-1$ and $\mathcal{K}_H, s_z, 0 \models \sigma'_2$ for all $z \in j..v$.

Therefore, $\mathcal{K}_H, s_a, 0 \models \mathsf{A}\Pi$.

Let $\sigma = \mathsf{E}\Pi \in \tau(s_a)$ be a (selectable) formula of the lowest priority. If $a < i$ then, by Proposition 46 ((c) and (d)), there exists non-empty $\Pi' \in \mathsf{Variants}(\Pi)$ such that $\mathsf{E}\Pi' \in \tau(n_i) \subseteq \tau(s_i)$ and

- For all $\Box(\sigma_1 \mathcal{U} \sigma_2) \in \Pi$: $\Box(\sigma_1 \mathcal{U} \sigma_2) \in \Pi'$ and for all $z \in a..i-1$: $\sigma_1 \in \tau(s_z)$ or $\sigma_2 \in \tau(s_z)$.
- For all $\Box(\sigma_1 \vee \Box\sigma_2) \in \Pi$: either $\Box(\sigma_1 \vee \Box\sigma_2) \in \Pi'$ and $\sigma_1 \in \tau(s_z)$ for all $z \in a..i-1$; or $\Box\sigma_2 \in \Pi'$ and there exists j such that $a \leq j \leq i$, so that for all $h \in a..j-1$: $\sigma_1 \in \tau(s_h)$ and for all $h \in j..i$: $\sigma_2 \in \tau(s_h)$.

Since $\mathsf{E}\Pi' \in \tau(n_1)$ is selected at some node n_h for some $i \leq h \leq j$, then by Proposition 48(c), $\sigma_2 \in \tau(s_w)$ for some $w \in i..j$. Moreover, by Proposition 46(c), for all $z \in i..j$: $\sigma_1 \in \tau(s_z)$ or $\sigma_2 \in \tau(s_z)$. In addition $\tau(n_h) = \Sigma_\ell$ for all $h \in i..j+1$, henceforth by Proposition 46(d), for all $\square(\sigma_1 \vee \square\sigma_2) \in \Pi'$: $\sigma_1 \in \tau(s_z)$ for all $z \in i..j$. Therefore, the application of the induction hypothesis to every σ_1 and every σ_2, allows us to ensure that $\mathcal{K}_H, s_a, 0 \models \mathsf{E}\Pi$. The case $a \geq i$ can be seen as the particular case where $\Pi' = \Pi$ and $\mathsf{E}\Pi \in \tau(n_h)$ for all $h \in i..j+1$.

Let $\sigma = \mathsf{A}\Pi \in \tau(s_a)$ be a (selectable) formula of the lowest priority. Then, Π contains at least one $\sigma_1 \,\mathcal{U}\, \square\sigma_2$ or $\square(\sigma_1 \,\mathcal{U}\, \sigma_2)$. We study two cases depending on whether there exists $\Pi' \in \mathsf{Variants}(\Pi)$ such that $\mathsf{A}\Pi' \in \Sigma_\ell$ or not. In the negative case, by Proposition 47 and the induction hypothesis, there exists $\sigma_1 \,\mathcal{U}\, (\sigma_2 \wedge \Diamond\sigma_3) \in \Pi$ and some j, j' such that $a \leq j \leq j' \leq k$, for all $h \in i..j-1$: $\mathcal{K}_H, s_h, 0 \models \sigma_1$, $\mathcal{K}_H, s_j, 0 \models \sigma_2$, and $\mathcal{K}_H, s_{j'}, 0 \models \sigma_3$. Therefore, $\mathcal{K}_H, s_a, 0 \models \mathsf{A}\Pi$. Otherwise, let $\mathsf{A}\Pi' \in \Sigma_\ell \subseteq \tau(s_i)$ such that $\Pi' \in \mathsf{Variants}(\Pi)$. By induction hypothesis and Proposition 47, $\mathcal{K}_H, s_i, 0 \models \mathsf{A}\Pi'$ suffices to ensure that $\mathcal{K}_H, s_a, 0 \models \mathsf{A}\Pi$. Hence, we are going to prove that $\mathcal{K}_H, s_i, 0 \models \mathsf{A}\Pi'$. If $\mathsf{A}\Pi'$ is non-selectable, then $\mathcal{K}_H, s_i, 0 \models \mathsf{A}\Pi'$ holds because $\mathcal{K}_H, s_h, 0 \models \sigma_1$ for all $\square(\sigma_1 \vee \square\sigma_2) \in \Pi'$ and all $h \in i..j$. Otherwise, $\mathsf{A}\Pi'$ is the lowest priority formula that is in the label of all the stages $s_i, s_{i+1}, \ldots, s_j$, hence for all $h \in i..j$:

- $\sigma_1 \in \tau(s_h)$ for all $\sigma_1 \,\mathcal{U}\, (\sigma_2 \wedge \Diamond\sigma_3) \in \Pi'$.
- $\{\sigma_1, \sigma_2\} \subseteq \tau(s_h)$ for all $\sigma_1 \,\mathcal{U}\, \square\sigma_2 \in \Pi'$.
- $\sigma_1 \in \tau(s_h)$ or $\sigma_2 \in \tau(s_h)$ for all $\square(\sigma_1 \,\mathcal{U}\, \sigma_2) \in \Pi'$.
- $\sigma_1 \in \tau(s_h)$ for all $\square(\sigma_1 \vee \square\sigma_2) \in \Pi'$.

Moreover, $\mathsf{A}\Pi'$ contains at least one $\sigma_1 \,\mathcal{U}\, \square\sigma_2$ or $\square(\sigma_1 \,\mathcal{U}\, \sigma_2)$, and one of its eventualities (that could be also of the form $\sigma_1 \,\mathcal{U}\, (\sigma_2 \wedge \Diamond\sigma_3)$) is selected at some of the stages $s_{i_1}, s_{i_2}, \ldots, s_{i_z}$. Hence, Proposition 49, along with the induction hypothesis, ensure that $\mathcal{K}_H, s_i, 0 \models \mathsf{A}\Pi'$. ∎

Corollary 54. *For any expanded bunch H of $\mathcal{A}^{sys}_{\Sigma_0}$, $\mathcal{K}_H \models \Sigma_0$*

Proof. Immediate consequence of Lemma 53. ∎

Now, we prove the refutational completeness of the tableau method.

Theorem 55 (Refutational Completeness). *For any set of state formulae Σ_0, if $\mathsf{UnSat}(\Sigma_0)$ then there exists a closed tableau for Σ_0.*

Proof. Suppose the contrary, that there exists no closed tableau for Σ_0. Then the systematic tableau $\mathcal{A}^{sys}_{\Sigma_0}$ would be open and there would be at least one expanded bunch H in $\mathcal{A}^{sys}_{\Sigma_0}$. By Corollary 54, $\mathcal{K}_B \models \Sigma_0$. Consequently Σ_0 would be satisfiable. ∎

7.4 Termination

Most tableau systems for modal and temporal logics, satisfy the *analytic super-formula property (ASP)*: for every finite set of formulae Σ, there exists a finite set that contains all the formulae that may occur in any tableau for Σ. Such a set is usually called the closure of Σ. The ASP also ensures the non-existence of infinite branches where all the nodes have different labels. Hence, by controlling loops, the finiteness of proof search can be ensured. In our case, as a consequence of the β^+-rules, the tableau system fails to satisfy the ASP, but it satisfies a slightly weaker variant which ensures completeness and that we call the *weak analytic superformula property (WASP)*: for every finite set of state formulae Σ_0 there exists a finite set (usually called the local closure of Σ) that contains all the formulae that may occur in any (systematic) tableau for Σ constructed by Algorithm $\mathcal{A}^{sys}$. For this purpose, the eventuality selection policy used in the $\mathcal{A}^{sys}$ is crucial.

Theorem 56 (Termination of the Tableau Method). *For any set of state formulae Σ_0 , the construction of the expanded tableau $\mathcal{A}^{sys}_{\Sigma_0}$ terminates.*

Proof. Tableau rules produce a finite branching, hence König's Lemma, applies. The subsumption-based simplification rules (Subsection 3.5) do prevent the generation of formulae containing the original eventuality when a "new variant" has been generated. By Propositions 48 and 49, the application of a β^+-rule to a selected formula stops after a finite number of steps. Finally, Proposition 52 ensures that any open branch is eventually-covered. ∎

Theorem 57 (Completeness of the Tableau Method). *For any set of state formulae Σ_0, if Σ_0 is satisfiable then there exists a (finite) open expanded tableau for Σ_0.*

Proof. The existence of the systematic tableau $\mathcal{A}^{sys}_{\Sigma_0}$ suffices to prove this fact, by Theorem 56. ∎

8 Complexity

The satisfiability problems for CTL and CTL* are known to be EXPTIME-complete [11] and 2EXPTIME-complete [26] (see also [20]). Our logic $\mathsf{ECTL}^{\#}$ lies between CTL and CTL*. In this section, we show that the size of our systematic tableau $\mathcal{A}^{sys}_{\Sigma}$ for an initial set of state-formulae Σ of size m is bounded by $2^{2^{\mathcal{O}(m^2)}}$.

The tree-style tableau has two types of nodes: the AND-nodes –roots of bunches where the $(\mathsf{Q}\circ)$-rule applies– and the others (OR-nodes). Any subtree whose root is an AND-node and whose leaves are either closed or AND-nodes corresponds to an iteration of the main loop in Algorithm 1. The leaves to which the $(\mathsf{Q}\circ)$-rule applies are elementary. A single iteration of Algorithm 1, performs at most once application of a β^+-rule to the marked eventuality, along with (at most) one application of an α- or β-rule for each path subformula different from the marked eventuality.

We first calculate a bound for the number of nodes in the subtree generated by an iteration of the systematic tableau (Algorithm 1). To this end, we utilize the set $\mathsf{Comp}(\Sigma)$ formed by the union of all the $\alpha\beta^+$-saturated sets that contains Σ (see Definition 40). The application of the α, β, and β^+-rules generates at most 3 children, except in the case of $(\mathsf{A}\sigma)$-rule. But w.l.o.g. we can see the $(\mathsf{A}\sigma)$-rule application to $\mathsf{A}(\sigma_1, \sigma_2, \cdots, \sigma_m, \pi)$ as m applications of a binary simpler rule producing $S_{\beta_1} = \{\sigma_1\}, S_{\beta_2} = \{\mathsf{A}(\sigma_2, \cdots, \sigma_m, \pi)\}$. Then $S_{\beta_{2,1}} = \{\sigma_2\}, S_{\beta_{2,2}} = \{\mathsf{A}(\sigma_3, \cdots, \sigma_m, \pi)\}$ and so on. Henceforth, assuming some natural notion of size, $|\_|$ for a set of state formulae (for example, the number of symbols in the set), we can bound the number of nodes in a single iteration, as follows.

Proposition 58. *Let Ψ be any uniform set of state formulae such that $|\Psi| = m$. The number of nodes generated, in the construction of $\mathcal{A}^{sys}_{\Psi}$, by single iteration of the Algorithm 1, until every leaf is labelled by an elementary set (see Definition 12), is bounded by $2^{2^{\mathcal{O}(m)}}$.*

Proof. The depth of the sub-tree generated by an iteration of $\mathcal{A}^{sys}_{\Psi}$ is bounded by $|\mathsf{Comp}(\Psi)|$. Adding the $\circ$ symbol to the initial set of m symbols in Ψ, the number of formulae in $\mathsf{Comp}(\Psi)$ is bounded by the number of different subformulae constructed with $m+1$ symbols. Hence, $|\mathsf{Comp}(\Psi)| \leq 2^{m+1}$. Since the width of the tableau is bounded by 3 the total number of possible nodes is bounded by $3^{2^{m+1}} \leq 2^{2^{m+2}}$, which is bounded by $2^{2^{\mathcal{O}(m)}}$. ∎

To complete one iteration of the Algorithm 1, in a tree of elementary leaves, the rule $(\mathsf{Q}\circ)$ is applied to every elementary leaf. The number of nodes produced by each application of $(\mathsf{Q}\circ)$ depends on the number of formulae of the form $\mathsf{E}\Pi$ (E-formulae).

Proposition 59. *Let Ψ be any uniform set of state formulae such that $|\Psi| = m$. The number of E-formulae, in an elementary leaf of $\mathcal{A}^{sys}_{\Psi}$, after a single iteration of the Algorithm 1, is bounded by $\mathcal{O}(m^2)$.*

Proof. The worst case is when Ψ contains E-formulae with nested eventualities. The construction of one of the branches requires to unfold all nested eventualities, as follows

$$\tau(l) = \{\mathsf{E}(\mathsf{E}(\cdots(\mathsf{E}(\mathsf{E}(\mathsf{E}(\sigma_1 \,\mathcal{U}\, \Box\sigma_2)\,\mathcal{U}\, \Box\sigma_3)\,\mathcal{U}\, \Box\sigma_4 \cdots)\,\mathcal{U}\, \Box\sigma_{m-1})\,\mathcal{U}\, \Box\sigma_m), \cdots\} \tag{2}$$

The successive application of $(\mathsf{E}\,\mathcal{U}\,\Box)$-rule produces the next set.

$$\begin{array}{ll} \{ & \mathsf{E}\circ(\cdots(\mathsf{E}(\mathsf{E}(\mathsf{E}(\sigma_1 \,\mathcal{U}\, \Box\sigma_2)\,\mathcal{U}\, \Box\sigma_3 \cdots)\,\mathcal{U}\, \Box\sigma_m), \mathsf{E}\circ(\cdots(\mathsf{E}(\mathsf{E}(\mathsf{E}(\sigma_1 \,\mathcal{U}\, \Box\sigma_2)\,\mathcal{U}\, \Box\sigma_3 \cdots)\,\mathcal{U}\, \Box\sigma_{m-1}), \\ & \mathsf{E}\circ(\cdots(\mathsf{E}(\mathsf{E}(\mathsf{E}(\sigma_1 \,\mathcal{U}\, \Box\sigma_2)\,\mathcal{U}\, \Box\sigma_3 \cdots)\,\mathcal{U}\, \Box\sigma_{m-2}), \mathsf{E}\circ(\cdots(\mathsf{E}(\mathsf{E}(\mathsf{E}(\sigma_1 \,\mathcal{U}\, \Box\sigma_2)\,\mathcal{U}\, \Box\sigma_3 \cdots)\,\mathcal{U}\, \Box\sigma_{m-3}), \\ & \vdots \\ & \mathsf{E}\circ(\mathsf{E}(\mathsf{E}(\mathsf{E}(\sigma_1 \,\mathcal{U}\, \Box\sigma_2)\,\mathcal{U}\, \Box\sigma_3)\,\mathcal{U}\, \Box\sigma_4)\,\mathcal{U}\, \Box\sigma_5), \mathsf{E}\circ(\mathsf{E}(\mathsf{E}(\sigma_1 \,\mathcal{U}\, \Box\sigma_2)\,\mathcal{U}\, \Box\sigma_3)\,\mathcal{U}\, \Box\sigma_4), \\ & \mathsf{E}\circ(\mathsf{E}(\sigma_1 \,\mathcal{U}\, \Box\sigma_2)\,\mathcal{U}\, \Box\sigma_3), \mathsf{E}\circ(\sigma_1 \,\mathcal{U}\, \Box\sigma_2), \sigma_1, \cdots \quad \} \end{array}$$

Unfolding the $(m-1)$ eventualities produces $m-1$ new formulae and so on. In general, when there are m nested eventualities in Ψ, the number of E-formulae increases in $m + (m-1) + (m-2) + .. + 1$, which is of $\mathcal{O}(m^2)$. ∎

Proposition 60. *Let Ψ be any uniform set of state formulae such that $|\Psi| = m$. The number of nodes (labelled by uniform sets) in $\mathcal{A}_{\Psi}^{sys}$, after a single iteration of the Algorithm 1, is bounded by $2^{2^{\mathcal{O}(m)}}$.* ∎

Proof. By Proposition 58, there is $2^{2^{\mathcal{O}(m)}}$ possible elementary leaves. By Proposition 59 each elementary leaf has at most $\mathcal{O}(m^2)$ E-formulae. Hence, any of these leaves might split into $\mathcal{O}(m^2)$ new nodes by the application of the $(\mathsf{Q}\circ)$-rule. So far, $(\mathcal{O}(m^2) \times 2^{2^{\mathcal{O}(m)}})$ nodes. Hence, of order is $2^{2^{\mathcal{O}(m)}}$. ∎

Next, we give an upper bound of the number of iterations the algorithm executes while an eventuality is kept marked. At the end of an iteration, after the application of $(\mathsf{Q}\circ)$-rule and the simplifications rules, either there is a marked eventuality or, otherwise, Algorithm 1 marks one. Next propositions are useful to show that although, in the worst case, the next-step variants of the marked eventuality increases the size of the eventuality, such growing is bounded by $2^{\mathcal{O}(m^2)}$.

Proposition 61. *Let l be a non-terminal leaf with a marked eventuality such that $|\tau(l)| = m$. Assume one iteration has been executed on l, and that l' is a non-terminal leaf after that iteration such that $\tau(l')$ contains the marked eventuality. Then, the size of $\tau(l')$ except for the marked eventuality is bounded by $\mathcal{O}(m^2)$.*

Proof. It is easy to see that the application of rules $(\mathsf{A}\Box\mathcal{U})$ and $(\mathsf{E}\Box\mathcal{U})$ cause an exponential growing of the number of formulae in $\tau(l)$. However, after the application of the $(\mathsf{Q}\circ)$-rule and the simplifications rules $(\sqsubset\mathsf{Q}\Box\mathcal{U})$, $(\sqsubset\mathsf{Q}\sigma\mathcal{U})$, $(\sqsubset\mathsf{Q}\mathcal{U}\sigma)$, the subsumed formulae are removed. Hence $\tau(l')$ remains bounded by m. Consequently, the worst case is when the rule $(\mathsf{Q}\mathcal{U}\sigma)$ (and similarly for $(\mathsf{Q}\mathcal{U}\Box)$) is applied as in the equation (2). Then, as in the proof of Proposition 59, the size of the set of formulae $\tau(l')$ except for the marked eventuality can increases in at most $\mathcal{O}(m^2)$. ∎

Proposition 62. *Let l be a non-terminal leaf with a marked eventuality and let $|\tau(l)| = m$. The number of iterations the systematic tableau executes along a branch whose initial node is l, while this marked eventuality is kept marked, is bounded by $2^{\mathcal{O}(m^2)}$.*

Proof. In each iteration, a next-step variant of the marked eventuality is generated. Once this is done, it remains without changing until the next iteration. By Proposition 61, when the β^+-rule applies to the marked eventuality, the size of the node (except for the marked eventuality) is bounded by $\mathcal{O}(m^2)$. Henceforth, after $2^{\mathcal{O}(m^2)}$ iterations, the next-step variants make that any open branch should contain a loop (though could be eventuality-covered or not). ∎

Finally, we give an upper bound on the number of times that an eventuality could be marked.

Proposition 63. *Let l be any non-terminal leaf in the systematic tableau such that $\tau(l)$ contains m eventualities. Each eventuality in $\tau(l)$ is marked at most m times, along a branch of the systematic tableau construction.*

Proof. The proof is based on some notions introduced in the proof of Proposition 52. Hence, we recall that a cycle-uniform set is exclusively formed by non-selectable and lowest priority formulae, and for Σ and Σ' being any two cycle-uniform sets of state formulae, we say that $\Sigma' \preceq \Sigma$ if, and only if, every formula in Σ' is either a proper subformula of some formula $\mathsf{Q}\Pi \in \Sigma$ or its negation; or a formula $\mathsf{Q}\Pi'$ such that there exist $\mathsf{Q}\Pi \in \Sigma$ such that $\Pi' \in \mathsf{Variants}(\Pi)$. Moreover, for any cycle-uniform set Σ, $\preceq$ is a well-founded order on the collection of all cycle-uniform sets Σ' such that $\Sigma' \preceq \Sigma$.

By Propositions 50 and 51, each eventuality in a highest priority formula is selected at most once before a cycle-uniform set is got in any open branch of the systematic tableau construction. Hence, the result is trivially true for the eventualities inside highest priority formulae in $\tau(\ell)$ that are marked before the first state that is labelled by a cycle-uniform set. Hence, we concentrate on the case that $\tau(\ell)$ is a cycle-uniform set that contains m eventualities. For $m = 0$ the result is trivial. For $m > 0$, each of the m eventualities must occur inside a lowest priority formula. In the worst case, after marking each of the m eventualities, the systematic procedure gets a non-terminal leaf ℓ' such that $\tau(\ell') \prec \tau(\ell)$ and $\tau(\ell')$ contains $m-1$ eventualities. By induction hypothesis, each of the latter is marked $m-1$ times. Therefore, adding the previous mark the latter $m-1$ eventualities are marked m times along the branch (whereas the unique eventuality in $\tau(\ell) \setminus \tau(\ell')$ is marked only once). ∎

Proposition 64. *Let Σ be a set of state-formulae and let m be the size of Σ. The systematic tableau $\mathcal{A}_{\Sigma}^{sys}$ has at most $2^{2^{\mathcal{O}(m^2)}}$ nodes.*

Proof. Each iteration in the construction of $\mathcal{A}_{\Sigma}^{sys}$ has at most a marked eventuality. Obviously the number of eventualities is bounded by m. Therefore, by Proposition 63, the whole process makes m^2 marks. By Proposition 62, the number of iterations along a branch is bounded by $m \times 2^{\mathcal{O}(m^2)}$, hence $2^{\mathcal{O}(m^2)}$. Since, by Proposition 60, the number of nodes of each iteration is bounded by $2^{2^{\mathcal{O}(m)}}$, this is $(2^{2^{\mathcal{O}(m)}})^{2^{\mathcal{O}(m^2)}}$. Therefore, $\mathcal{A}_{\Sigma}^{sys}$ has at most $2^{2^{\mathcal{O}(m^2)}}$ nodes. ∎

9 Conclusion

We introduced a new logic, $\mathsf{ECTL}^{\#}$, in the family of BTL and its tree-style one pass tableau. This extends the expressiveness of fairness by a new class of fairness constraints with the $\mathcal{U}$ operator. We presented a tree-style one pass tableaux method for this logic and proved its correctness. The tableau method handles inputs in an 'analytic" way, due to the new, crucial for branching structures, concept of 'inner context', in which eventualities are to be fulfilled. The tableau rules that invoke the inner context, are essential to handle A-disjunctive formulae. Our analysis of A-disjunctive and E-conjunctive formulae and of the prioritisation of eventualities, based on their structure and the context for their fulfillment, are important from the methodological point of view.

We note that the size of the systematic tableau for the input of size m is bounded by $2^{2^{O(m^2)}}$ (see Section 8). However, the method aims at the 'shortest' way to fulfil the eventualities and, for many examples, finds the first open bunch, giving us a model for the tableau input. This significantly reduces the complexity.

The presented technique is amenable for implementation – and this will be another stream of our future work. In the refinement and implementation of the algorithm we will be able to rely on similar techniques used in the implementation of its linear-time analogue ([13]).

Our tableau technique is not directly extensible to $\mathsf{CTL}^{\star}$. Without any significant modifications, β^{+}-rules become unsound for inputs that are beyond $\mathsf{ECTL}^{\#}$ syntax due to nested path subformulae as in $\mathsf{A}\Diamond(\circ p \wedge \mathsf{E}\circ\neg p)$ mentioned in Figure 1. In order to illustrate this issue consider the following generalization of the above example.

Example 65. Let $\Sigma, \mathsf{A}\Diamond((\circ\sigma_1) \wedge \sigma_2)$ where $\Sigma, \sigma_1, \sigma_2$ is a set of state formulae. We try to apply a new rule of the β^{+}-style to this set of state formulae. Then, we should obtain two children: $\Sigma, \mathsf{A}\circ\sigma_1, \sigma_2$ and $\Sigma, \mathsf{A}\circ((\neg\Sigma)\,\mathcal{U}\,((\circ\sigma_1) \wedge \sigma_2))$. For soundness, the unsatisfiability of both children must ensure the unsatisfiability of the initial set. However, supposing that any model of Σ is neither a model of $(\mathsf{A}\circ\sigma_1) \wedge \sigma_2$, nor a model of $\mathsf{A}\circ((\neg\Sigma)\,\mathcal{U}\,((\circ\sigma_1) \wedge \sigma_2))$, we cannot ensure that a countermodel of $\mathsf{A}\Diamond((\circ\sigma_1) \wedge \sigma_2)$ can be constructed. In fact, in one of the cases we should consider that exists $\mathcal{K}$ and two paths $x, y \in \mathsf{fullpaths}(K)$ such that $\mathcal{K}, x \models \circ\neg\sigma_1$ and $\mathcal{K}, y \models \circ\neg((\neg\Sigma)\,\mathcal{U}\,((\circ\sigma_1) \wedge \sigma_2))$. Even in the case that the latter would be held because $\mathcal{K}, y \models \circ\Box((\circ\neg\sigma_1) \vee \neg\sigma_2)$, we cannot ensure that $\mathcal{K}, y \models (\circ\neg\sigma_1) \vee \neg\sigma_2$. Indeed, the path y could satisfies $\circ\sigma_1$ and also σ_2 according to our assumptions. This fact also prevents the construction of a limit path starting with a segment of the path y (whose last state satisfies Σ) is neither possible.

It is worthy to note that $\mathsf{A}\Diamond((\Diamond\sigma_1) \wedge \sigma_2)$ is an $\mathsf{ECTL}^{\#}$-formula with applicable β^{+}-rules. The interested reader could simplify the construction in the ('if' direction) proof for the rule $(\mathsf{A}\,\mathcal{U}\,\sigma)^{+}$ in Lemma 37 to the case of $\mathsf{A}\Diamond((\Diamond\sigma_1) \wedge \sigma_2)$; and compare it with the above unsuccessful construction of a countermodel of $\mathsf{A}\Diamond((\circ\sigma_1) \wedge \sigma_2)$.

We would like to remark that applying the unsound rule discussed above (and, after, our rules) to the unsatisfiable singleton of the $\mathsf{CTL}^{\star}$-formula $\mathsf{A}\Diamond(\circ p \wedge \mathsf{E}\circ\neg p)$ produces (by chance) a closed tableau. However, the unsoundness of that rule makes that the tableau is also closed for the satisfiable singleton of the $\mathsf{CTL}^{\star}$-formula $\mathsf{A}\Diamond(\circ\circ p \wedge \mathsf{E}\circ\circ\neg p)$. ∎

In spite of the difficulties explained in the previous example, for the proof of correctness of β^{+}-rules, we developed the technique to identify relevant state formulae inside the specific path-modalities. This technique will be useful in studying more expressive logics (e.g. $\mathsf{CTL}^{\star}$), as it allows to identify those subformulae that do not affect the 'context', thus enabling the simplification of the structures.

References

1. Pietro Abate, Rajeev Goré, and Florian Widmann. One-pass tableaux for computation tree logic. In Nachum Dershowitz and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning*, pages 32–46, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
2. Therese Berg and Harald Raffelt. Model checking. In Manfred Broy, Bengt Jonsson, Joost-Pieter Katoen, Martin Leucker, and Alexander Pretschner, editors, *Model-Based Testing of Reactive Systems*, pages 557–603. Springer-Verlag, Berlin Heidelberg, 2005.
3. Alexander Bolotov, Montserrat Hermo, and Paqui Lucio. Extending Fairness Expressibility of ECTL+: A Tree-Style One-Pass Tableau Approach. In Natasha Alechina, Kjetil Nørvåg, and Wojciech Penczek, editors, *25th International Symposium on Temporal Representation and Reasoning (TIME 2018)*, volume 120 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:22. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018.
4. Kai Brünnler and Martin Lange. Cut-free sequent systems for temporal logic. *Journal of Logic and Algebraic Programming*, 76(2):216–225, 2008.
5. Serenella Cerrito, AmÃľlie David, and Valentin Goranko. Optimal tableau method for constructive satisfiability testing and model synthesis in the alternating-time temporal logic ATL+. volume 17, 2014.
6. Edmund M. Clarke, E. Allen Emerson, and Aravinda P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Trans. Program. Lang. Syst.*, 8(2):244–263, 1986.
7. E. Allen Emerson. Temporal and modal logic. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science (Vol. B)*, pages 995–1072. MIT Press, Cambridge, USA, 1990.
8. E. Allen Emerson and Joseph Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *Journal of Computer and System Sciences*, 30(1):1 – 24, 1985.
9. E. Allen Emerson and Joseph Y. Halpern. Sometimes and not never revisited: On branching versus linear time temporal logic. *J. ACM*, 33(1):151–178, 1986.
10. E. Allen Emerson and Chin-Laung Lei. Temporal reasoning under generalized fairness constraints. In Monien B. and Vidal-Naquet G., editors, *STACS 1986, Lecture Notes in Computer Science, vol 210*, pages 21–37. Springer-Verlag Berlin Heidelberg, Karlsruhe, Federal Republic of Germany, 1986.
11. Michael J. Fischer and Richard E. Ladner. Propositional modal logic of programs. In *Proceedings of the Ninth Annual ACM Symposium on Theory of Computing*, STOC '77, pages 286–294, New York, NY, USA, 1977. ACM.
12. Oliver Friedmann, Markus Latte, and Martin Lange. A decision procedure for CTL* based on tableaux and automata. In Jürgen Giesl and Reiner Hähnle, editors, *Automated Reasoning*, pages 331–345, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
13. Jose Gaintzarain, Montserrat Hermo, Paqui Lucio, Marisa Navarro, and Fernando Orejas. Dual systems of tableaux and sequents for PLTL. *Journal of Logic and Algebraic Programming*, 78(8):701–722, 2009.
14. Rajeev Gore. Tableau methods for modal and temporal logics. In Marcello D'Agostino, Dov M. Dov Gabbay, Reiner Hähnle, and Joachim Posegga, editors, *Handbook of Tableau Methods*, pages 297–396. Springer, Netherlands, Dordrecht, 1999.
15. Bernhard Josko. Model checking of ctl formulae under liveness assumptions. In Thomas Ottmann, editor, *Automata, Languages and Programming, 14th International Colloquium*, pages 5–24. Springer-Verlag Berlin Heidelberg, Karlsruhe, Federal Republic of Germany, 1987.
16. Jan Kretinsky and Ruslan Ledesma Garza. Rabinizer 2: Small deterministic automata for LTL$\setminus$GU. In *Automated Technology for Verification and Analysis - 11th International Symposium, ATVA 2013*, pages 446–450, Heidelberg Dordrecht London New York, 2013. Springer.
17. Orna Kupferman, Moshe Y. Vardi, and Pierre Wolper. An automata-theoretic approach to branching-time model checking. *J. ACM*, 47(2):312–360, 2000.
18. John C. MaCabe-Dansted and Mark Reynolds. Rewrite rules for CTL*. *Journal of Applied Logic*, 21:24 – 56, 2017.
19. Nicolas Markey. Temporal logics. Course notes, Master Parisien de Recherche en Informatique, Paris, France, 2013.
20. Arne Meier, Martin Mundhenk, Michael Thomas, and Heribert Vollmer. The complexity of satisfiability for fragments of CTL and $\mathsf{CTL}^\star$. *Electronic Notes in Theoretical Computer Science*, 223:201 – 213, 2008. Proceedings of the Second Workshop on Reachability Problems in Computational Models (RP 2008).
21. Amir Pnueli and Yonit Kesten. A deductive proof system for CTL*. In Luboš Brim, Mojmír Křetínský, Antonín Kučera, and Petr Jančar, editors, *CONCUR 2002 — Concurrency Theory*, pages 24–40, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.
22. Mark 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.

23. Mark Reynolds. A tableau-based decision procedure for CTL*. *Formal Aspects of Computing*, 23(6):739–779, 2011.
24. Robert S. Streett and E. Allen Emerson. The propositional mu-calculus is elementary. In Jan Paredaens, editor, *Automata, Languages and Programming, 11th Colloquium, Antwerp, Belgium, July 16-20, 1984, Proceedings*, volume 172 of *Lecture Notes in Computer Science*, pages 465–472. Springer, 1984.
25. Jun Sun, Yang Liu, Jin Song Dong, and Hai H. Wang. Specifying and verifying event-based fairness enhanced systems. In Shaoying Liu, Tom Maibaum, and Keijiro Araki, editors, *Formal Methods and Software Engineering: 10th International Conference on Formal Engineering Methods ICFEM 2008*, pages 5–24. Springer Science and Business Media, Kitakyushu-City, Japan, 2008.
26. M.Y. Vardi and L. Stockmeyer. Lower bound in full (2EXPTIME-hardness for CTL*-SAT). Online, available at `http://www.cs.rice.edu/~vardi/papers/ctl_star_lower_bound.pdf`, 1985.