CTL with Finitely Bounded Semantics*

Valentin Goranko[1], Antti Kuusisto[2], and Raine Rönnholm[3]

1 **Stockholm University, Stockholm, Sweden; and**
University of Johannesburg, Johannesburg, South Africa†
valentin.goranko@philosophy.su.se
2 **University of Bremen, Bremen, Germany**
kuusisto@uni-bremen.de
3 **University of Tampere, Tampere, Finland**
raine.ronnholm@uta.fi

Abstract

We consider a variation of the branching time logic CTL with non-standard, "finitely bounded" semantics (FBS). FBS is naturally defined as game-theoretic semantics where the proponent of truth of an eventuality must commit to a time limit (number of transition steps) within which the formula should become true on all (resp. some) paths starting from the state where the formula is evaluated. The resulting version $\mathrm{CTL}_{\mathbf{FB}}$ of CTL differs essentially from the standard one as it no longer has the finite model property.

We develop two tableaux systems for $\mathrm{CTL}_{\mathbf{FB}}$. The first one deals with infinite sets of formulae, whereas the second one deals with finite sets of formulae in a slightly extended language allowing explicit indication of time limits in formulae. We prove soundness and completeness of both systems and also show that the latter tableaux system provides an EXPTIME decision procedure for it and thus prove EXPTIME-completeness of the satisfiability problem.

1998 ACM Subject Classification F.4.1 Mathematical Logic, I.2.4 Knowledge Representation Formalisms and Methods

Keywords and phrases CTL, finitely bounded semantics, tableaux, decidability

Digital Object Identifier 10.4230/LIPIcs.TIME.2017.14

1 Introduction

The branching time logic CTL ([4]) is interpreted in transition systems and its language involves quantification over all infinite paths (computations) starting at the current state. It is well known that in order to determine truth of any CTL formula in a *finite* transition system, it suffices to consider only sufficiently long finite prefixes of paths starting at the current state, in the sense that any existential (resp. universal) eventuality is satisfied at that state iff it is satisfied on some (resp. all) of these finite prefixes. Furthermore, satisfiability/validity in CTL has the finite model property. It is natural, therefore, to consider a finitary version of the semantics of CTL, restricted on all models to finite paths. Such a version has recently emerged as a compositional counterpart of the *finitely bounded game-theoretic semantics* (FBS), developed for the multi-agent extension ATL of CTL in [6] (see also [7] for an extended and revised version), where the length of the evaluation game is constrained by finite time limits which the players must set and decrease after every transition move in the game.

* The work of Valentin Goranko was supported by a research grant 2015-04388 of the Swedish Research Council. The work of Antti Kuusisto was supported by the ERC grant 647289 "CODA."
† Visiting professorship.

Intuitively, the main distinctive feature of the FBS is that a player who claims an eventuality formula to be true must commit to a time bound (number of transition steps) within which she can defend that claim by fulfilling that eventuality. Technically, the FBS for CTL can be obtained by changing the definition of temporal operators so that a uniform bound on the number of transition steps needed to fulfill a given eventuality is imposed.

In this paper we study the alternative version $\mathrm{CTL}_{\mathbf{FB}}$ of CTL, defined by the FBS. $\mathrm{CTL}_{\mathbf{FB}}$ differs essentially from CTL with standard semantics. In particular, it falsifies the fundamental fixed-point characterizations of the operators EG and AU. Based on this we show that $\mathrm{CTL}_{\mathbf{FB}}$ lacks the finite model property and that the set of validities of $\mathrm{CTL}_{\mathbf{FB}}$ is a proper subset of the validities of CTL.

We develop two equivalent versions of tableaux for $\mathrm{CTL}_{\mathbf{FB}}$. The first tableaux deals with infinite sets of CTL formulae, while the second one involves only finite sets of formulae, but in a suitably extended language which allows explicit indication of *time limits* by corresponding *indexing parameters*. Essentially, the parameters enable encoding infinitary formulae by finite ones. We prove soundness and completeness of both tableau systems and also show that the latter system provides a decision procedure for $\mathrm{CTL}_{\mathbf{FB}}$. We thereby establish that satisfiability in that logic is decidable and EXPTIME-complete.

We note that even though not being the actual motivation for the present study, a major independent justification for considering bounded semantics for CTL is (the conceptual idea behind) *bounded model checking* [2], which provides the main link with related previous work. Other versions of bounded semantics for CTL, based on evaluation of formulae on finite paths, have been considered in the context of bounded model checking in e.g. [8], [9], [10].

2 Preliminaries: CTL with finitely bounded semantics

Here we only provide brief preliminaries on CTL. For further details see e.g. [5, Ch.7].

2.1 The standard semantics of CTL

▶ **Definition 1.** A **transition system** is a tuple $\mathcal{T} = (\mathrm{S}, \mathrm{R})$, where S is a nonempty set of **states** and $\mathrm{R} \subseteq \mathrm{S} \times \mathrm{S}$ a **transition relation**. We also assume that R is *serial*, i.e. for every $s \in \mathrm{S}$ there is $s' \in \mathrm{S}$ such that $(s, s') \in \mathrm{R}$. A **path** in $\mathcal{T}$ is a sequence $\lambda : \mathbb{N} \to \mathrm{S}$ of states such that $(\lambda(n), \lambda(n{+}1)) \in \mathrm{R}$ for every $n \in \mathbb{N}$.

An **interpreted transition system** (ITS) over $\mathcal{T}$ is a tuple $\mathcal{M} = (\mathrm{S}, \mathrm{R}, \Phi, \ell)$, where Φ a set of **proposition symbols** and $\ell : \mathrm{S} \to \mathcal{P}(\Phi)$ is a **state description function** defining for every state s the set of atomic propositions true at that state.

The syntax CTL is given by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{EX}\,\varphi \mid \mathsf{E}(\varphi\,\mathsf{U}\,\varphi) \mid \mathsf{A}(\varphi\,\mathsf{U}\,\varphi)\,.$$

We also use the following abbreviations $\mathsf{AX}\,\varphi := \neg\mathsf{EX}\,\neg\varphi$, $\mathsf{EF}\,\varphi := \mathsf{E}(\top\,\mathsf{U}\,\varphi)$, $\mathsf{AF}\,\varphi := \mathsf{A}(\top\,\mathsf{U}\,\varphi)$, $\mathsf{EG}\,\varphi := \neg\mathsf{AF}\,\neg\varphi$ and $\mathsf{AG}\,\varphi := \neg\mathsf{EF}\,\neg\varphi$.

▶ **Definition 2.** Let $\mathcal{M} = (\mathrm{S}, \mathrm{R}, \Phi, \ell)$ be an interpreted transition system, $s \in \mathrm{S}$ and φ a CTL-formula. **Truth of φ at s in $\mathcal{M}$**, denoted by $\mathcal{M}, s \models \varphi$, is defined as follows:

- $\mathcal{M}, s \models p$ iff $p \in \ell(s)$.
- $\mathcal{M}, s \models \neg\varphi$ iff $\mathcal{M}, s \not\models \varphi$.
- $\mathcal{M}, s \models \varphi \vee \psi$ iff $\mathcal{M}, s \models \varphi$ or $\mathcal{M}, s \models \psi$.
- $\mathcal{M}, s \models \mathsf{EX}\,\varphi$ iff there is a state s' such that $(s, s') \in \mathrm{R}$ and $\mathcal{M}, s' \models \varphi$.

- $\mathcal{M}, s \models \mathsf{E}(\varphi \,\mathsf{U}\, \psi)$ iff there is a path λ starting from s and $i \geq 0$ such that $\mathcal{M}, \lambda(i) \models \psi$ and $\mathcal{M}, \lambda(j) \models \varphi$ for every $j < i$.
- $\mathcal{M}, s \models \mathsf{A}(\varphi \,\mathsf{U}\, \psi)$ iff for every path λ starting from s, there is $i \geq 0$ such that $\mathcal{M}, \lambda(i) \models \psi$ and $\mathcal{M}, \lambda(j) \models \varphi$ for every $j < i$.

We define the following operators on formulae, where $\mathsf{Q} \in \{\mathsf{E}, \mathsf{A}\}$:

$$\mathbf{G}_{\mathsf{Q};\theta}(\varphi) := \theta \wedge \mathsf{QX}\varphi; \quad \mathbf{U}_{\mathsf{Q};\psi,\theta}(\varphi) := \theta \vee (\psi \wedge \mathsf{QX}\varphi)\,.$$

It is well-known that, for each $\mathsf{Q} \in \{\mathsf{E}, \mathsf{A}\}$, in terms of their semantics:

- $\mathsf{QG}\,\theta$ is the greatest fixpoint of the operator $\mathbf{G}_{\mathsf{Q};\theta}$, i.e. $\mathsf{QG}\,\theta \equiv \nu Z.\mathbf{G}_{\mathsf{Q};\theta}(Z)$,
- $\mathsf{Q}(\psi \,\mathsf{U}\, \theta)$ the least fixpoint of the operator $\mathbf{U}_{\mathsf{Q};\psi,\theta}$, i.e. $\mathsf{Q}(\psi \,\mathsf{U}\, \theta) \equiv \mu Z.\mathbf{U}_{\mathsf{Q};\psi,\theta}(Z)$.

Now, we define recursively on $n \in \mathbb{N}$ the respective iterations of these operators:

- $\mathbf{G}^0_{\mathsf{Q}}(\theta) := \theta$; $\mathbf{G}^{n+1}_{\mathsf{Q}}(\theta) := \mathbf{G}_{\mathsf{Q};\theta}(\mathbf{G}^n_{\mathsf{Q}}(\theta))$
- $\mathbf{U}^0_{\mathsf{Q}}(\psi, \theta) := \theta$; $\mathbf{U}^{n+1}_{\mathsf{Q}}(\psi, \theta) := \mathbf{U}_{\mathsf{Q};\psi,\theta}(\mathbf{U}^n_{\mathsf{Q}}(\psi, \theta))$.

2.2 Game-Theoretic semantics for CTL

In [7] we have defined game-theoretic semantics (GTS) for the alternating time temporal logic ATL ([1], [5, Ch.9]) by using *evaluation games* between two players, Abelard and Eloise. Since CTL can be regarded as a (1-agent) fragment of ATL, a simple GTS for CTL can be obtained from the mentioned GTS for ATL in a straightforward (but not immediate) way.

▶ **Definition 3.** Let $\mathcal{M} = (\mathrm{S}, \mathrm{R}, \Phi, \ell)$ be an ITS, $s_{in} \in \mathrm{S}$ and φ a CTL-formula. The **(unbounded) evaluation game** $\mathcal{G}(\mathcal{M}, s_{in}, \varphi)$ is defined as follows. A **position** of the game is a tuple $(\mathbf{P}, s, \psi)$ where $\mathbf{P} \in \{\mathrm{Abelard}, \mathrm{Eloise}\}$, $s \in \mathrm{S}$ and ψ is a subformula of φ. The **initial position** of the game is $(\mathrm{Eloise}, s_{in}, \varphi)$. The evaluation game proceeds according to the following rules.

1. A position of the form $(\mathbf{P}, s, p)$, where $p \in \Phi$, is called an **ending position**. If $p \in \ell(s)$, then $\mathbf{P}$ wins the evaluation game. Else the **opposing player of P**, denoted by $\overline{\mathbf{P}}$, wins.
2. In $(\mathbf{P}, s, \neg\psi)$ the game moves to the next position $(\overline{\mathbf{P}}, s, \psi)$.
3. In $(\mathbf{P}, s, \psi \vee \theta)$ the player $\mathbf{P}$ decides whether the next position is $(\mathbf{P}, s, \psi)$ or $(\mathbf{P}, s, \theta)$.
4. In $(\mathbf{P}, s, \mathsf{EX}\,\psi)$ the player $\mathbf{P}$ may choose any state s' such that $(s, s') \in \mathrm{R}$ and the next position is $(\mathbf{P}, s', \psi)$.

For the rules for the formulae $\mathsf{E}(\psi \,\mathsf{U}\, \theta)$ and $\mathsf{A}(\psi \,\mathsf{U}\, \theta)$, we define the **embedded game** $\mathbf{G} := \mathbf{g}(\mathbf{V}, \mathbf{L}, s_0, \psi_{\mathbf{V}}, \psi_{\overline{\mathbf{V}}})$, where $\mathbf{V}, \mathbf{L} \in \{Abelard, Eloise\}$, s_0 is a state, and $\psi_{\mathbf{V}}$ and $\psi_{\overline{\mathbf{V}}}$ are formulae. The player $\mathbf{V}$ is called the **verifier** (of the embedded game) and $\mathbf{L}$ the **leader**. These players may, but need not be, the same. We let $\overline{\mathbf{V}}$ and $\overline{\mathbf{L}}$ denote the opponents of $\mathbf{V}$ and $\mathbf{L}$, respectively. The embedded game $\mathbf{G}$ starts from the **initial state** s_0 and proceeds from any state s according to the following rules until an **exit position** is reached.

(i) $\mathbf{V}$ may end the game at the exit position $(\mathbf{V}, s, \psi_{\mathbf{V}})$.
(ii) $\overline{\mathbf{V}}$ may end the game at the exit position $(\mathbf{V}, s, \psi_{\overline{\mathbf{V}}})$.
(iii) $\mathbf{L}$ may select any state s' such that $(s, s') \in \mathrm{R}$ and then $\mathbf{G}$ is continued from s'.

If the embedded game $\mathbf{G}$ continues an infinite number of rounds, the verifier $\mathbf{V}$ loses the entire evaluation game. The rest of the rules for the evaluation game are defined as follows:

5. In $(\mathbf{P}, s, \mathsf{E}(\psi \,\mathsf{U}\, \theta))$ the game is continued from the exit position of $\mathbf{g}(\mathbf{P}, \mathbf{P}, s, \theta, \psi)$.
6. In $(\mathbf{P}, s, \mathsf{A}(\psi \,\mathsf{U}\, \theta))$ the game is continued from the exit position of $\mathbf{g}(\mathbf{P}, \overline{\mathbf{P}}, s, \theta, \psi)$.

In GTS, truth of a formula is defined as existence of a winning strategy for Eloise in the corresponding evaluation game. By [7] we obtain the following equivalence:

$$\mathcal{M}, s \models \varphi \text{ iff } \quad \text{Eloise has a winning strategy in } \mathcal{G}(\mathcal{M}, s, \varphi)\,.$$

2.3 Finitely bounded semantics for CTL

Note that unbounded evaluation games may continue infinitely long before a winner is determined. In order to avoid this, we have also defined **bounded evaluation games** for ATL in [7]. Such a game is obtained by modifying the unbounded game simply by attaching ordinal values, called **time limits**, to the embedded games. A time limit is announced by the verifier in the beginning of the embedded game and the verifier has to decrease it after every transition. Since ordinals are *well-founded*, it is guaranteed that the whole bounded evaluation game ends in a finite number of moves – even in infinite models.

If players are allowed to announce arbitrarily large time limits in the game, then unbounded and bounded evaluation games become equivalent[1]. A particularly interesting and natural variant of a bounded evaluation game is when only *finite* ordinals may be used by the players. We call the corresponding game-theoretic semantics **finitely bounded (FBS)** and denote its truth condition by $\models_{\text{fb}}$. As shown in [7], the FBS for ATL differs from the standard compositional semantics but corresponds to a natural variant of it, to be discussed further. In the special case of CTL, this semantics modifies only the truth conditions of AU and EU so that a uniform bound on the number of transition steps needed to fulfill a given eventuality is imposed.

(AU_{fb}) $\mathcal{M}, s \models_{\text{fb}} \mathsf{A}(\varphi \,\mathsf{U}\, \psi)$ iff there is $n \in \mathbb{N}$ such that for every path λ starting from s, there is $i \leq n$ such that $\mathcal{M}, \lambda(i) \models_{\text{fb}} \psi$ and $\mathcal{M}, \lambda(j) \models_{\text{fb}} \varphi$ for every $j < i$.

(EU_{fb}) $\mathcal{M}, s \models_{\text{fb}} \mathsf{E}(\varphi \,\mathsf{U}\, \psi)$ iff there is $n \in \mathbb{N}$, a path λ starting from s and $i \leq n$ such that $\mathcal{M}, \lambda(i) \models_{\text{fb}} \psi$ and $\mathcal{M}, \lambda(j) \models_{\text{fb}} \varphi$ for every $j < i$. (We note that since existential quantifiers commute, (EU_{fb}) is in fact equivalent to the standard truth definition of EU.)

Thus, the FBS of formulae of the type $\mathsf{E}(\varphi \,\mathsf{U}\, \psi)$, is standard, even though they will not be treated here as (existential) eventualities usually are, see Section 3.1.1. Furthermore, it is easy to see that the FBS given to $\mathsf{A}(\varphi \,\mathsf{U}\, \psi)$ by (AU_{fb}) is not equivalent to the standard one, and in fact, such formulae *do not behave like universal eventualities*, as they would in standard CTL. Respectively, the derived FBS for AG is equivalent to the standard one, while for EG we obtain the following non-equivalent version.

(EG_{fb}) $\mathcal{M}, s \models_{\text{fb}} \mathsf{EG}\,\varphi$ iff for every $n \in \mathbb{N}$, there is a path λ_n starting from s such that $\mathcal{M}, \lambda_n(i) \models_{\text{fb}} \varphi$ for every $i \leq n$. (Note that the path λ_n depends on n.)

By replacing the truth condition for AU (and EG) with the ones above, we obtain CTL **with finitely bounded semantics**, denoted by $\text{CTL}_{\mathbf{FB}}$.

For a set of formulae Γ, by $\mathcal{M}, s \models_{\text{fb}} \Gamma$ we denote the claim that $\mathcal{M}, s \models_{\text{fb}} \varphi$ for all $\varphi \in \Gamma$. Satisfiability and validity of $\text{CTL}_{\mathbf{FB}}$ formulae are defined and denoted as usual.

2.4 Some properties of CTL_{FB}

All observations made for the finitely bounded semantics of ATL in [7] apply directly to $\text{CTL}_{\mathbf{FB}}$. Here are the most important and interesting ones.

1. On all *image finite models* (where every state has finitely many immediate successors) $\text{CTL}_{\mathbf{FB}} = \text{CTL}$, i.e., truth of CTL-formulae is independent of which semantics is used.
2. $\text{CTL} \neq \text{CTL}_{\mathbf{FB}}$ in models that have infinite branchings. In particular, the fixed point properties of the operators F and G fail since the implications $\mathsf{EG}\,p \to (p \wedge \mathsf{EX}\,\mathsf{EG}\,p)$ and (dually) $(p \vee \mathsf{AX}\,\mathsf{AF}\,p) \to \mathsf{AF}\,p$ are valid in CTL but not in $\text{CTL}_{\mathbf{FB}}$. For an explicit model where the former implication fails, see the ITS in Figure 4 in Section 3.4.

[1] It suffices to use ordinals that have the same cardinality as the model. For more details, see [7].

3. Since CTL has the finite model property, the two facts above imply that $\mathrm{CTL}_{\mathbf{FB}}$ *does not have the finite model property*, as these implications cannot fail in (image-)finite models. It is thus not immediately obvious that satisfiability for $\mathrm{CTL}_{\mathbf{FB}}$ should be decidable.
4. Consequently, the set of validities of $\mathrm{CTL}_{\mathbf{FB}}$ is properly included in the set of validities of CTL. Indeed, every non-validity of CTL is falsified in a finite model and thus, by fact 1, it is a non-validity of $\mathrm{CTL}_{\mathbf{FB}}$, too.

On the one hand, the lack of finite model property of $\mathrm{CTL}_{\mathbf{FB}}$ can be regarded as an increase of the semantic complexity and richness in comparison with standard CTL. But on the other hand, the semantics of $\mathrm{CTL}_{\mathbf{FB}}$ can be seen as simpler than that of CTL, in the sense that it only requires one to consider finite paths and does not involve dealing with universal eventualities.

Note the conceptual parallels between FBS and *for-loops* on the one side, and between the standard semantics and *while-loops* on the other. For more on this, see Section 5.2 of [7].

Finally, we list in the lemma below some validities in $\mathrm{CTL}_{\mathbf{FB}}$ used further. These can be verified easily by using the respective fixpoint characterizations.

▶ **Lemma 4.** *For every ITS $\mathcal{M}$ and $s \in \mathcal{M}$ the following hold, for $\mathsf{Q} \in \{\mathsf{E}, \mathsf{A}\}$:*
1. $\mathcal{M}, s \models_{\mathrm{fb}} \mathsf{QG}\,\varphi$ *iff* $\mathcal{M}, s \models_{\mathrm{fb}} \mathbf{G}^n_{\mathsf{Q}}(\varphi)$ *for every* $n \in \mathbb{N}$.
2. $\mathcal{M}, s \models_{\mathrm{fb}} \mathsf{Q}(\varphi\,\mathsf{U}\,\psi)$ *iff* $\mathcal{M}, s \models_{\mathrm{fb}} \mathbf{U}^n_{\mathsf{Q}}(\varphi, \psi)$ *for some* $n \in \mathbb{N}$.
3. $\models_{\mathrm{fb}} \mathsf{AG}\,\varphi \to \mathbf{G}^n_{\mathsf{A}}(\varphi)$ *and* $\models_{\mathrm{fb}} \mathbf{U}^n_{\mathsf{E}}(\varphi, \psi) \to \mathsf{E}(\varphi\,\mathsf{U}\,\psi)$ *for every* $n \in \mathbb{N}$.
4. $\models_{\mathrm{fb}} \mathbf{G}^n_{A}(\theta) \to \mathbf{G}^m_{A}(\theta)$ *and* $\models_{\mathrm{fb}} \mathbf{U}^m_{A}(\psi, \theta) \to \mathbf{U}^n_{A}(\psi, \theta)$ *for all* $m, n \in \mathbb{N}$ *such that* $m < n$.

3 Infinitary tableaux for $\mathrm{CTL}_{\mathbf{FB}}$

We only provide a detailed sketch of the (infinitary) tableaux-building procedure for $\mathrm{CTL}_{\mathbf{FB}}$ here. For further details that are essentially the same as in the tableaux method for the standard CTL, see [5, Chapter 13], the style of which we closely follow here.

3.1 Preliminaries

3.1.1 Types and components of formulae

In this section we will regard each of $\top, \bot, \neg, \wedge, \mathsf{EX}, \mathsf{AX}, \mathsf{EG}, \mathsf{AG}, \mathsf{EU}, \mathsf{AU}$ as primitive connectives in the language, while $\vee, \to, \leftrightarrow, \mathsf{EF}, \mathsf{AF}$ will be regarded as abbreviations. We will distinguish five *types of formulae*: **literals, conjunctive, disjunctive, existential successor and universal successor** formulae. Literals are atomic propositions and negations of these, and $\top$, $\bot$. Successor formulae are those beginning with EX (existential) and AX (universal). For each of the latter three types of formulae listed above we define their respective **components** as in Figure 1. Literals have no components. We write $scomp(\chi)$ to denote the successor component of the formula χ. Note that formulae of the type EU and $\neg\mathsf{AG}$, even though having standard semantics, are not treated here as existential eventualities are treated in standard tableaux for CTL. This is mainly for the sake of uniformity with the finitary tableaux for $\mathrm{CTL}_{\mathbf{FB}}$ presented further.

▶ **Lemma 5.** *For every ITS $\mathcal{M}$, $s \in \mathcal{M}$ and a formula φ of* $\mathrm{CTL}_{\boldsymbol{FB}}$ *the following hold:*
1. *If φ is any conjunctive formula, then $\mathcal{M}, s \models_{\mathrm{fb}} \varphi$ iff $\mathcal{M}, s \models_{\mathrm{fb}} \psi$ for every conjunctive component ψ of φ.*
2. *If φ is any disjunctive formula, then $\mathcal{M}, s \models_{\mathrm{fb}} \varphi$ iff $\mathcal{M}, s \models_{\mathrm{fb}} \psi$ for some disjunctive component ψ of φ.*

successor formula	successor component	conjunctive formula	conjunctive components	disjunctive formula	disjunctive components
$\mathsf{EX}\,\varphi$ (exist.)	φ	$\neg\neg\varphi$	φ		
$\mathsf{AX}\,\varphi$ (univ.)	φ	$\varphi \wedge \psi$	$\varphi,\ \psi$	$\neg(\varphi \wedge \psi)$	$\neg\varphi,\ \neg\psi$
$\neg\mathsf{AX}\,\varphi$ (exist.)	$\neg\varphi$	$\mathsf{AG}\,\varphi$	$\{\varphi, \mathsf{AX}\,\mathsf{AG}\,\varphi\}$	$\neg\mathsf{AG}\,\varphi$	$\{\neg\mathbf{G}^n_{\mathsf{A}}(\varphi)\}_{n\in\mathbb{N}}$
$\neg\mathsf{EX}\,\varphi$ (univ.)	$\neg\varphi$	$\mathsf{EG}\,\varphi$	$\{\mathbf{G}^n_{\mathsf{E}}(\varphi)\}_{n\in\mathbb{N}}$	$\neg\mathsf{EG}\,\varphi$	$\{\neg\mathbf{G}^n_{\mathsf{E}}(\varphi)\}_{n\in\mathbb{N}}$
		$\neg\mathsf{E}(\varphi\,\mathsf{U}\,\psi)$	$\{\neg\psi, \neg\varphi \vee \neg\mathsf{EX}\,\mathsf{E}(\varphi\,\mathsf{U}\,\psi)\}$	$\mathsf{E}(\varphi\,\mathsf{U}\,\psi)$	$\{\mathbf{U}^n_{\mathsf{E}}(\varphi,\psi)\}_{n\in\mathbb{N}}$
		$\neg\mathsf{A}(\varphi\,\mathsf{U}\,\psi)$	$\{\neg\mathbf{U}^n_{\mathsf{A}}(\varphi,\psi)\}_{n\in\mathbb{N}}$	$\mathsf{A}(\varphi\,\mathsf{U}\,\psi)$	$\{\mathbf{U}^n_{\mathsf{A}}(\varphi,\psi)\}_{n\in\mathbb{N}}$

Figure 1 Types and components of formulae in $\mathrm{CTL}_{\mathbf{FB}}$.

3.1.2 Extended closure of a formula

In order to determine the truth of a $\mathrm{CTL}_{\mathbf{FB}}$ formula η one has to consider a set of 'simpler' formulae which appear in the process of the truth evaluation of η and are needed in the tableaux construction for it. As in the case of CTL, some of these auxiliary formulae are not really simpler, as they come from the unfoldings of the temporal operators and are generally not subformulae of η. Still, they can be identified quite simply and collected in the **extended closure** of the formula η, denoted $ecl(\eta)$, obtained by closing under taking respective components of already added formulae, which is defined generically as follows.

▶ **Definition 6.** The **extended closure** $ecl(\varphi)$ of formula φ is the least set of formulae such that $\varphi \in ecl(\varphi)$ and $ecl(\varphi)$ is closed under taking all conjunctive, disjunctive, successor components of formulae in $ecl(\varphi)$. For any set of formulae Γ we define $ecl(\Gamma) := \bigcup\{\, ecl(\varphi) \mid \varphi \in \Gamma \,\}$. A set of formulae Γ is **closed** if $\Gamma = ecl(\Gamma)$.

▶ **Example 7.** Let $\eta := \mathsf{EG}\,p \wedge \neg(p \wedge \mathsf{EX}\,\mathsf{EG}\,p)$. This formula will be used in the running examples later on. Here is the extended closure of η: $ecl(\eta) = \{\eta, \mathsf{EG}\,p, \neg(p \wedge \mathsf{EX}\,\mathsf{EG}\,p)\} \cup \{\mathbf{G}^n_{\mathsf{E}}(p)\}_{n\in\mathbb{N}} \cup \{\neg p, \neg\mathsf{EX}\,\mathsf{EG}\,p\} \cup \{\mathsf{EX}\,\mathbf{G}^n_{\mathsf{E}}(p)\}_{n\in\mathbb{N}} \cup \{\neg\mathsf{EG}\,p\} \cup \{\neg\mathsf{EX}\,\mathbf{G}^n_{\mathsf{E}}(p)\}_{n\in\mathbb{N}} \cup \{\neg\mathbf{G}^n_{\mathsf{E}}(p)\}_{n\in\mathbb{N}}$.

3.1.3 Full expansions

▶ **Definition 8.** A set of formulae is **patently inconsistent** if it contains $\bot$, or $\neg\top$, or a contradictory pair of formulae φ and $\neg\varphi$.

▶ **Definition 9.** A set of formulae Γ is **fully expanded** iff:

1. it is not patently inconsistent,
2. for every conjunctive formula in Γ, all of its conjunctive components are in Γ,
3. for every disjunctive formula in Γ, at least one of its disjunctive components is in Γ.

▶ **Definition 10.** A fully expanded set of formulae Ψ is a **full expansion** of a set of formulae Γ if Ψ can be obtained from Γ by the following procedure FullExpansion, consisting in repeated application of the following rules, where initially no formula is marked as 'used':

(C-comp) for every conjunctive formula φ in the current set Γ', not yet marked as 'used', add all of its conjunctive components to Γ' and mark φ as 'used'.

(D-comp) for every disjunctive formula φ in the current set Γ', not yet marked as 'used', add to Γ' one of its disjunctive components that is not already in Γ' and mark φ as 'used'.

Since the procedure FullExpansion is non-deterministic, it can produce (possibly infinitely) many full expansions of Γ (or none). This procedure can be readily replaced with a deterministic process that constructs all fully expanded sets allowed by FullExpansion.

Moreover, in the finitary tableau (see Section 4), this process runs in EXPTIME. We denote the set of all full expansions of Γ (obtained by the procedure FullExpansion) by $\mathrm{FE}(\Gamma)$. Intuitively, a full expansion of Γ consists of all formulae appearing on some open branch in the saturated local (propositional) tableau for input set Γ.

▶ **Proposition 11.** *For any set of* $\mathrm{CTL}_{\boldsymbol{FB}}$*-formulae* Γ*, ITS* $\mathcal{M}$ *and state* $s \in \mathcal{M}$*:*

$$\mathcal{M}, s \models_{\mathrm{fb}} \Gamma \text{ iff } \mathcal{M}, s \models_{\mathrm{fb}} \Delta \text{ for some } \Delta \in \mathrm{FE}(\Gamma)\,.$$

The proof for this proposition is routine, using Lemma 5.

The purpose of the tableaux method outlined further is to determine whether at least one full expansion of the input formula set is satisfiable.

▶ **Example 12.** Let $\Gamma_1 := \{\eta\}$, where $\eta := \mathsf{EG}\,p \wedge \neg(p \wedge \mathsf{EX}\,\mathsf{EG}\,p)$ (recall Example 7) and let $\Gamma_2 := \{\mathbf{G}^k_{\mathsf{E}}(p), \neg\mathsf{EG}\,p\}$, where $k \in \mathbb{N}$. These sets of formulae will be used later in Example 17.

Γ_1 has the following full expansion: $\{\eta, \mathsf{EG}\,p, \neg(p \wedge \mathsf{EX}\,\mathsf{EG}\,p)\} \cup \{\mathbf{G}^n_{\mathsf{E}}(p)\}_{n\in\mathbb{N}} \cup \{\neg\mathsf{EX}\,\mathsf{EG}\,p\} \cup \{\mathsf{EX}\,\mathbf{G}^n_{\mathsf{E}}(p)\}_{n\in\mathbb{N}}$. This is the only full expansion of Γ_1, because choosing the disjunctive component $\neg p$ of $\neg(p \wedge \mathsf{EX}\,\mathsf{EG}\,p)$ creates a patently inconsistent set (since $\mathbf{G}^0_{\mathsf{E}}(p) = p$).

On the other hand, the set $\mathrm{FE}(\Gamma_2)$ is infinite, because for each $m \in \mathbb{N}$ such that $m \notin \{0, k\}$ the set $\Psi^m = \{\mathbf{G}^k_{\mathsf{E}}(p), \neg\mathsf{EG}\,p,\ p,\ \mathsf{EX}\,\mathbf{G}^{k-1}_{\mathsf{E}}(p), \neg\mathbf{G}^m_{\mathsf{E}}(p), \neg\mathsf{EX}\,\mathbf{G}^{m-1}_{\mathsf{E}}(p)\}$ is a full expansion of Γ_2. The disjunctive components $\neg\mathbf{G}^k_{\mathsf{E}}(p)$ and $\neg\mathbf{G}^0_{\mathsf{E}}(p) = \neg p$ of $\neg\mathsf{EG}\,p$ are the only ones that create patently inconsistent sets.

3.2 Hintikka structures

Intuitively, a Hintikka structure represents a partly defined rooted ITS satisfying the input formula. It is a graph, every node of which is labeled by a set of formulae. These labels are fully expanded subsets of the extended closure of a designated input formula, the satisfiability of which is tested by the tableau. All desired properties of the transition relations in a Hintikka structure are encoded by means of the labels of the states. Membership to the label of the state of a Hintikka structure simulates the notion of truth of a formula at a state of an interpreted transition system and the labelling of states must ensure that the Hintikka structure can generate a model of the input formula. The purpose of the tableau construction is to check for existence of a Hintikka structure 'satisfying' the input formula, in the sense described above. Here is the formal definition of a Hintikka structure for $\mathrm{CTL}_{\mathbf{FB}}$.

▶ **Definition 13.** Given a closed set of $\mathrm{CTL}_{\mathbf{FB}}$-formulae Γ, a **Hintikka structure (HS) for** Γ is a tuple $\mathcal{H} = (\mathrm{S}, \mathrm{R}, \mathrm{H})$ s.t. (S, R) is a transition system and $\mathrm{H} : \mathrm{S} \to \mathcal{P}(\Gamma)$ is a labelling function satisfying the following conditions for every $s \in \mathrm{S}$:

(H1) $\mathrm{H}(s)$ is fully expanded (in the sense of Def. 9).

(H2) If $\varphi \in \mathrm{H}(s)$ is an existential successor formula, then $scomp(\varphi) \in \mathrm{H}(s')$ for some s' such that $s\,\mathrm{R}\,s'$. (Recall that $scomp(\varphi)$ denotes the successor component of the formula φ.)

(H3) If $\varphi \in \mathrm{H}(s)$ is a universal successor formula, then $scomp(\varphi) \in \mathrm{H}(s')$ for every s' such that $s\,\mathrm{R}\,s'$.

▶ **Definition 14.** A formula $\varphi \in \mathrm{CTL}_{\mathbf{FB}}$ is **satisfiable in a Hintikka structure** $\mathcal{H} = (\mathrm{S}, \mathrm{R}, \mathrm{H})$ for a set Γ if $\varphi \in \mathrm{H}(s)$ for some $s \in \mathrm{S}$. A set of formulae $\Psi \subseteq \Gamma$ is **satisfiable in** $\mathcal{H}$ if $\Psi \subseteq \mathrm{H}(s)$ for some state s in $\mathcal{H}$.

Note that every rooted ITS uniformly generates a rooted Hintikka structure for any closed set of formulae Γ by labelling each state of the ITS with the set of all formulae from Γ that are true at that state. Formally:

▶ **Lemma 15.** *For any closed set of* $\mathrm{CTL}_{\boldsymbol{FB}}$*-formulae* Γ *(in the sense of Def. 6) over as set of atomic propositions* Φ *and for every interpreted transition system* $\mathcal{M} = (\mathrm{S}, \mathrm{R}, \Phi, \ell)$ *the structure* $\mathcal{H}(\mathcal{M}) = (\mathrm{S}, \mathrm{R}, \mathrm{H})$*, where* $\mathrm{H}(s) = \{\varphi \in \Gamma \mid \mathcal{M}, s \models \varphi\}$ *for every* $s \in \mathrm{S}$*, is a Hintikka structure for* Γ*.*

An essential difference between interpreted transition systems and Hintikka structures is that, while an ITS determines the truth value of every formula at every state, a Hintikka structure only contains just enough information to determine the truth values of those formulae that are directly involved in the evaluation of the input formula η at the root state.

Given a formula φ for which we look for a model, we will be interested in Hintikka structures for the set $ecl(\varphi)$. For that class of Hintikka structures to be suitable for our purpose, every formula satisfiable in a Hintikka structure must also be satisfiable in an ITS, so the two notions of satisfiability are equivalent. Thus, the following result is needed. (See the Appendix for a sketch of proof).

▶ **Theorem 16.** *A* $\mathrm{CTL}_{\boldsymbol{FB}}$*-formula* η *is satisfiable iff it is satisfiable in some Hintikka structure for* $ecl(\eta)$*.*

3.3 Construction of the tableaux

The tableau procedure presented here attempts to construct for a given input formula η a non-empty graph $\mathcal{T}^\eta$, called a **tableau**, representing (in a way) sufficiently many possible Hintikka structures for η. The procedure consists of three major phases:

1. **Construction phase.** In that phase a finite directed graph $\mathcal{P}^\eta$ with labeled vertices, called the **pretableau** for η, is produced, following prescribed **construction rules**. The set of nodes of the pretableau properly contains the set of nodes of the tableau $\mathcal{T}^\eta$ that is to be ultimately built. The pretableau has two types of nodes: **states** and **prestates**. The states are labeled with fully expanded subsets of $ecl(\eta)$ and represent states of a Hintikka structure, while the prestates can be labeled with any subsets of $ecl(\eta)$ and they play only an auxiliary and temporary role. In this tableau construction states and prestates with an already existing label are not created again, but reused. So, when there is no danger of confusion, we will identify prestates or states with their labels.
2. **Prestate elimination phase.** Here the prestates are removed using the **prestate elimination rule**. The result is a smaller graph $\mathcal{T}_0^\eta$, called the **initial tableau** for η.
3. **State elimination phase.** In this phase we remove, using **state elimination rules**, all (if any) states from $\mathcal{T}_0^\eta$ that cannot be satisfied in any Hintikka structure. In the case of $\mathrm{CTL}_{\mathbf{FB}}$, where there are no explicit eventualities to be satisfied, an elimination of a state can happen for only one reason: some of the successor states that the state needs for the satisfaction of its successor formulae, have already turned out unsatisfiable and have been removed in the elimination process so far. The state elimination phase results in a (possibly empty) subgraph $\mathcal{T}^\eta$ of $\mathcal{T}_0^\eta$, called the **final tableau** for η.

If there is a state in the final tableau $\mathcal{T}^\eta$ containing η in its label, the tableau is declared **open** and the input formula η is pronounced satisfiable; otherwise, the tableau is declared **closed** and η is pronounced unsatisfiable.

3.3.1 The pretableau construction phase

The pretableau construction phase consists of two rules: $\textsc{PrExp}^{\mathrm{CTL}_{\mathbf{FB}}}$ producing all **offspring states** of a given prestate; and $\textsc{Next}^{\mathrm{CTL}_{\mathbf{FB}}}$ producing the **successor prestates** of

a given state. The rule PrExp$^{\mathrm{CTL}_{\mathbf{FB}}}$ involves the procedure FullExpansion, described in Section 3.1.3, for computing the family FE(Γ) of full expansions of a given set $\Gamma \subseteq ecl(\eta)$.

PrExp$^{\mathrm{CTL}_{\mathbf{FB}}}$: Given a prestate Γ to which the rule has not yet been applied, do the following:

1. Compute the family FE(Γ) of full expansions of Γ and add these as (labels of) new states in the pretableau, called the **offspring states** of Γ.
2. For each newly introduced state Δ, create an edge $\Gamma \dashrightarrow \Delta$.
3. If, however, the pretableau already contains a state with label Δ then do not create a new state with that label, but create an edge to Δ instead.

We denote the set $\{\Delta \mid \Gamma \dashrightarrow \Delta\}$ of offspring states of Γ by $\mathsf{states}(\Gamma)$. Hereafter we write $\mathsf{X}(\Delta) := \{\psi \mid \mathsf{AX}\psi \in \Delta\} \cup \{\neg\psi \mid \neg\mathsf{EX}\psi \in \Delta\}$ for any set of $\mathrm{CTL}_{\mathbf{FB}}$-formulae Δ.

Next$^{\mathrm{CTL}_{\mathbf{FB}}}$: Given a state Δ to which the rule has not yet been applied, do the following:

1. For each existential successor formula $\varphi \in \Delta$ (i.e., $\varphi = \mathsf{EX}\chi$ or $\varphi = \neg\mathsf{AX}\chi$) add a successor prestate Γ of Δ with label $\mathsf{X}(\Delta) \cup \{scomp(\varphi)\}$ and create an edge $\Delta \xrightarrow{\varphi} \Gamma$.
2. Consider the case where Δ has no existential successor formulae. If $\mathsf{X}(\Delta) \neq \emptyset$, add one prestate Γ with label $\mathsf{X}(\Delta)$ and an edge $\Delta \to \Gamma$. If $\mathsf{X}(\Delta) = \emptyset$, simply create a loop $\Delta \to \Delta$.
3. If the pretableau already contains a prestate with the label of Γ, then do not create a new prestate with that label, but create an edge to Γ instead.

The construction phase of building a pretableau for η begins with creating a single prestate $\{\eta\}$, followed by alternating applications of the rules PrExp$^{\mathrm{CTL}_{\mathbf{FB}}}$ and Next$^{\mathrm{CTL}_{\mathbf{FB}}}$, respectively to the prestates and the states created at the previous stages of the construction. The construction phase is completed if/when none of these rules can add any new states or prestates to the current graph. The resulting graph is the pretableau $\mathcal{P}^\eta$.

Note that there are two types of branching in the pretableau: **search branching**, from a prestate to its offspring states, indicated by $\dashrightarrow$, and **structural branching**, from a state to its successor prestates, indicated by $\xrightarrow{\chi}$ for an EX-formula χ. Search branching is branching of the search tree, and thus it is **disjunctive**, or **existential**: only one offspring state of every prestate is eventually needed to build a satisfying structure. Structural branching is **conjunctive**, or **universal**. Since it represents branching in the structure to be built, *all* successor prestates of every state are potentially needed in the construction.

In the subsequent figures illustrating tableau examples, prestates are indicated with shaded square boxes and labeled by P with indices, while states are indicated with transparent boxes with rounded corners and labeled by S with indices.

▶ **Example 17** (Pretableau). (Recall Example 12.) The sentence $\eta = \mathsf{EG}\,p \wedge \neg(p \wedge \mathsf{EX}\,\mathsf{EG}\,p)$ has the countably infinite pretableau given in Figure 2. From state S_0 there is an infinite structural branching to prestates $\{P_{0n} \mid n \in \mathbb{N}\}$. Since the pretableau construction is analogous for all of these prestates, we only present a single one of them here, namely P_{0k}. From P_{0k} there is an infinite search branching to states $\{S_{0km} \mid m \in \mathbb{N} \setminus \{0, k\}\}$. But all cases where $0 < m < k$ are analogous to each other. Similarly, all cases where $m > k$ are analogous to each other. Note that when $m < k$, the prestates $P_{0km0^{m-1}}$ do not have any offspring states since then $\mathrm{FE}(\{\mathbf{G}_{\mathsf{E}}^{k-m}(p), \neg\mathbf{G}_{\mathsf{E}}^{0}(p)\}) = \emptyset$.

3.3.2 Prestate elimination phase and initial tableaux

In this phase, all prestates are removed from $\mathcal{P}^\eta$, together with their incoming and outgoing arrows, by applying the following generic **prestate elimination rule**:

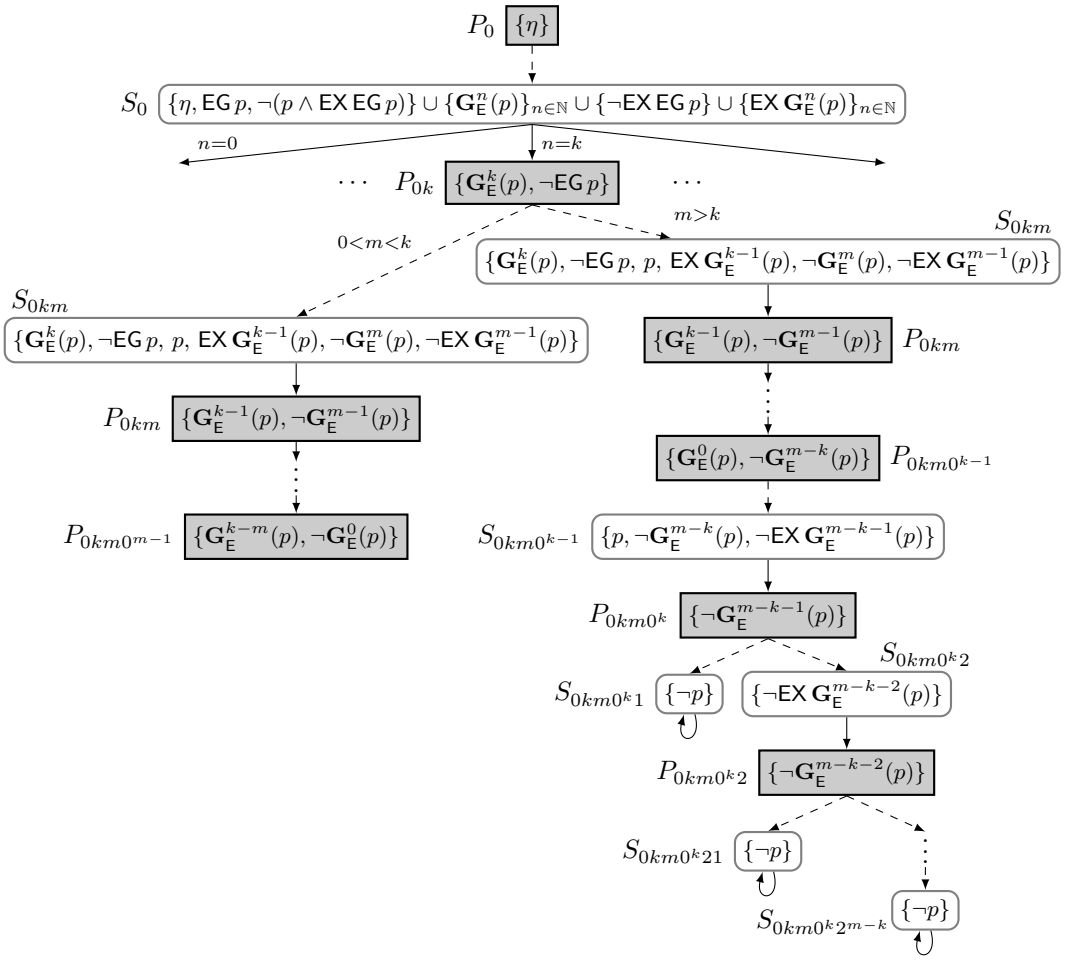

Figure 2 Infinitary pretableau for $\eta = \mathsf{EG}\,p \wedge \neg(p \wedge \mathsf{EX}\,\mathsf{EG}\,p)$.

PrestateElim$^{\mathrm{CTL}_{\mathrm{FB}}}$. For every prestate Γ in $\mathcal{P}^\eta$, do the following:

1. If there is a $\Delta \in \mathcal{P}^\eta$ with $\Delta \to \Gamma$, then for every $\Delta' \in \mathsf{states}(\Gamma)$, create an edge $\Delta \to \Delta'$;
2. Remove Γ from $\mathcal{P}^\eta$ together with its outgoing arrows.

The resulting graph is called the **initial tableau** for η, denoted $\mathcal{T}^\eta_0$. The offspring states of the input prestate $\{\eta\}$ are called **input states** of $\mathcal{T}^\eta_0$.

3.3.3 State elimination phase and the final tableau

This phase is carried out in a sequence of stages, starting with the initial tableau $\mathcal{T}^\eta_0$, and eliminating at each stage n at least one state for the current tableau $\mathcal{T}^\eta_n$, by applying the state elimination rule stated below, to produce the new current tableau $\mathcal{T}^\eta_{n+1}$, until stabilisation.

StateElim$^{\mathrm{CTL}_{\mathrm{FB}}}$: If a state Δ, containing an existential successor formula $\mathsf{EX}\,\psi$ (respectively, $\neg\mathsf{AX}\,\psi$), has no successor states containing ψ (respectively, $\neg\psi$) in the current tableau, then remove Δ from the tableau. Remove Δ from the tableau also if Δ has no successor states.

The rule StateElim$^{\mathrm{CTL}_{\mathrm{FB}}}$ is applied repeatedly until reaching a stage when no further elimination of states is possible. Such a stage may not be reached at any finite stage, so the elimination phase may have to go on for transfinitely many steps, in which case every limit-stage tableau is the limit of the decreasing chain (by inclusion) of current tableaux.

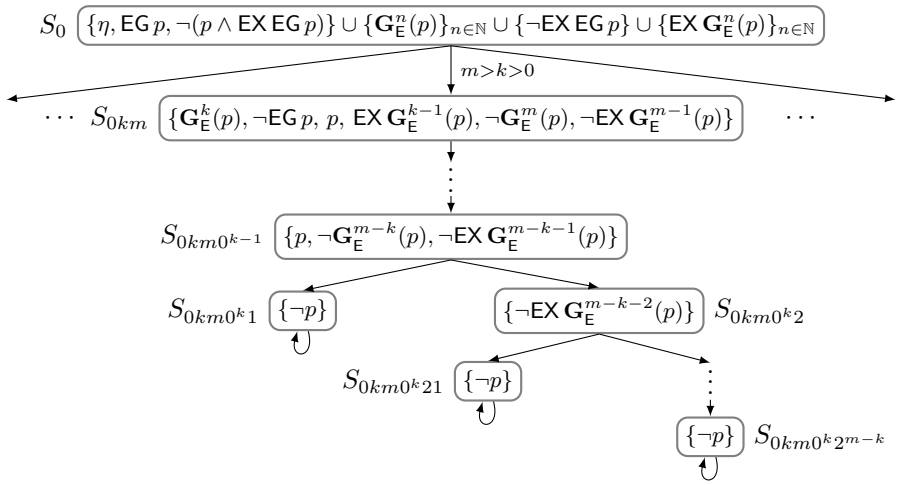

Figure 3 Final tableau for $\eta = \mathsf{EG}\, p \wedge \neg(p \wedge \mathsf{EX}\, \mathsf{EG}\, p)$.

When the state elimination phase reaches a (finite or limit) stabilization stage, the resulting tableau then is the **final tableau for** η, denoted by $\mathcal{T}^\eta$, with a set of states denoted by S^η.

▶ **Definition 18.** The final tableau $\mathcal{T}^\eta$ is **open** if $\eta \in \Delta$ for some $\Delta \in S^\eta$; else, $\mathcal{T}^\eta$ is **closed**.

The tableau procedure returns "not satisfiable" if the final tableau is closed; otherwise, it returns "satisfiable" and, moreover, provides sufficient information for producing a countable Hintikka structure satisfying η, as described in the completeness proof below.

▶ **Example 19** (Final tableau)**.** Recall the pretableau from Example 17. The initial tableau is obtained removing the prestates by applying $\textsc{PrestateElim}^{\mathrm{CTL}_{\mathbf{FB}}}$. The branches starting from states S_{0km} where $0 < m < k$ end at a prestate that has no offspring state. Therefore all of states on these branches are eliminated when the final tableau is formed. All the other states remain in the final tableau and therefore it is open. See Figure 3 for the final tableau.

3.4 Soundness and completeness of the infinitary tableau for $\mathrm{CTL}_{\mathbf{FB}}$

Note that in general none of the three phases of the construction terminates at any finite stage if performed consecutively, because the set of full expansions of a prestate may be infinite, or because infinitely many successor prestates may have to be introduced. However, each of these stages does reach saturation at some transfinite stage.

Soundness of the tableau procedure means that if the input formula η is satisfiable, then the final tableau $\mathcal{T}^\eta$ is open, so the tableau procedure is guaranteed to establish the satisfiability. A generic proof of soundness consists of showing that at least one tableau state containing η will survive forever, i.e. will never be eliminated. Note first that if η is satisfiable, and hence propositionally consistent, there will be at least one offspring state of the initial prestate containing η, due to Proposition 11. Moreover, for every rooted ITS $(\mathcal{M}, s)$ satisfying η, the set $\{\psi \in \mathit{ecl}(\eta) \mid \mathcal{M}, s \models \psi\}$ contains at least one such state. Thereafter, it suffices to show that *only unsatisfiable states* get eliminated in the state elimination phase. Thus, we will establish the soundness of the tableau for $\mathrm{CTL}_{\mathbf{FB}}$ by proving that the state elimination rule is 'sound' in a sense that it never eliminates a state with a satisfiable label. The soundness of the overall procedure is then an immediate consequence. The proof of that

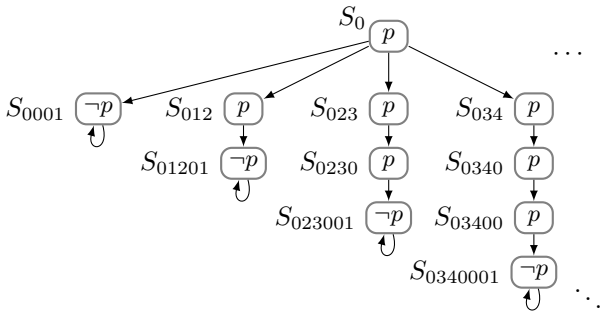

Figure 4 A model satisfying $\mathsf{EG}\, p \wedge \neg(p \wedge \mathsf{EX}\, \mathsf{EG}\, p)$.

claim follows a fairly routine argument (see [5, Chapter 13]) by induction on the applications of the state elimination rule. However, since the elimination phase for $\mathrm{CTL}_{\mathbf{FB}}$ may take a transfinite sequence of steps, the soundless argument now uses transfinite induction. For a proof sketch, see the Appendix.

▶ **Lemma 20.** *Let Γ be a prestate of $\mathcal{P}^\eta$ such that $\mathcal{M}, s \models \Gamma$ for some rooted interpreted transition system $(\mathcal{M}, s)$. Then $\mathcal{M}, s \models \Psi$ for at least one $\Psi \in \mathsf{states}(\Gamma)$.*

▶ **Lemma 21.** *No satisfiable state $\Delta \in \mathcal{T}_0^\eta$ is removed by any application of* $\textsc{StateElim}^{\mathrm{CTL}_{FB}}$ *during the state elimination phase.*

▶ **Theorem 22** (Soundness)**.** *If $\eta \in \mathrm{CTL}_{\boldsymbol{FB}}$ is satisfiable then $\mathcal{T}^\eta$ is open.*

Completeness of the tableau procedure means that if the input formula η is not satisfiable, then the final tableau $\mathcal{T}^\eta$ is closed, so the tableau procedure is guaranteed to establish the unsatisfiability. By contraposition, completeness means that if the final tableau is open, the input formula η is satisfiable. The proof of the completeness is sketched in the Appendix.

▶ **Theorem 23** (Completeness)**.** *For any formula $\eta \in \mathrm{CTL}_{\boldsymbol{FB}}$, if the final tableau $\mathcal{T}^\eta$ is open, then η is satisfiable.*

▶ **Example 24.** Since the final tableau of $\eta = \mathsf{EG}\, p \wedge \neg(p \wedge \mathsf{EX}\, \mathsf{EG}\, p)$ is open, η is satisfiable by Theorem 23. Thus $\mathsf{EG}\, p \to (p \wedge \mathsf{EX}\, \mathsf{EG}\, p)$ is not a valid $\mathrm{CTL}_{\mathbf{FB}}$ sentence. Consider the final tableau in Example 19 as a transition system $\mathcal{T}$. By using any state description function ℓ that satisfies the labels of the states in $\mathcal{T}$, we obtain an ITS which satisfies η.

However, by analysing the pretableau of η (Example 17) we can construct a simpler ITS that satisfies η. It suffices that, for every prestate Γ, a single state in $\mathsf{states}(\Gamma)$ survives the state elimination process. Thus, for a prestate P_{0k} ($k \in \mathbb{N}$), we can choose any offspring state S_{0km} for which $m > k$; e.g. we can choose $m := k + 1$. Furthermore, it suffices that, for a prestate P_{0km0^k}, we choose only the state S_{0km0^k1} (and remove the state S_{0km0^k2}).

By removing states from $\mathcal{T}^\eta$ as described above, we obtain the ITS in Figure 4 which can be seen as the simplest model satisfying η. Note that since $S_{0001}, S_{01201}, S_{023001}, S_{0340001}, \ldots$ have the same label $\{\neg p\}$, they are actually merged into a single state by the tableau rules.

▶ **Example 25.** Consider $\eta' := (p \wedge \mathsf{EX}\, \mathsf{EG}\, p) \wedge \neg \mathsf{EG}\, p$. Since $\neg \mathbf{G}^0_{\mathsf{E}}(p) = \neg p$ is a conjunctive component of $\neg \mathsf{EG}\, p$ and p is a conjunctive component of $p \wedge \mathsf{EX}\, \mathsf{EG}\, p$, we have $\mathrm{FE}(\{\eta'\}) = \emptyset$. Therefore the pretableau of η' will only have a single prestate, namely $\{\eta'\}$, and not any states. Thus the final tableau for η' is empty and thus it is trivially closed. Hence by Theorem 22 η' is not satisfiable, i.e. $(p \wedge \mathsf{EX}\, \mathsf{EG}\, p) \to \mathsf{EG}\, p$ is a valid $\mathrm{CTL}_{\mathbf{FB}}$ formula.

It is easy to show that the structural branchings in the tableau are always countable. Thus we can prove the following claim for $\mathrm{CTL}_{\mathbf{FB}}$. For more details, see the the Appendix.

▶ **Proposition 26.** $\mathrm{CTL}_{\boldsymbol{FB}}$ *has the* ***countable model property****, i.e. if* $\eta \in \mathrm{CTL}_{\boldsymbol{FB}}$ *is satisfiable, then* η *is satisfiable in a countable model.*

4 Finitary tableaux for $\mathrm{CTL}_{\mathsf{FB}}$

4.1 Construction of finitary tableaux

Hereafter we will use a set of new symbols $\{\mathbf{n}_i \mid i \in \mathbb{N}^+\}$, called **iteration parameters**, which will be used instead of natural numbers n_i in formulae of type $\mathbf{G}^{n_i}_{\mathsf{Q}}(\varphi)$ and $\mathbf{U}^{n_i}_{\mathsf{Q}}(\varphi,\psi)$. We denote the resulting extended language $\mathrm{CTL}^{par}_{\mathbf{FB}}$. Here $\mathbf{G}^{\mathbf{n}_i}_{\mathsf{Q}}(\varphi), \mathbf{U}^{\mathbf{n}_i}_{\mathsf{Q}}(\varphi,\psi) \in \mathrm{CTL}^{par}_{\mathbf{FB}}$ ($i \in \mathbb{N}^+$) *are not abbreviations*; they are treated as actual formulae of $\mathrm{CTL}^{par}_{\mathbf{FB}}$. We will not give any semantics for $\mathbf{G}^{\mathbf{n}_i}_{\mathsf{Q}}(\varphi)$ or $\mathbf{U}^{\mathbf{n}_i}_{\mathsf{Q}}(\varphi,\psi)$ as they will only be used "symbolically" in the construction of the finite tableau. In particular, the iteration parameter $\mathbf{n}_i$ is just a *symbol* which does not have any actual value. However, *intuitively*, each parameter $\mathbf{n}_i$ takes an "arbitrarily large" but finite and fixed value which represents the number of iterations of the given recursive operator. The index i (recall $i \in \mathbb{N}^+$) of the iteration parameter $\mathbf{n}_i$ indicates *when* the "value" of $\mathbf{n}_i$ has been fixed (with respect to the other iteration parameters). The "value" of $\mathbf{n}_i$ may depend on the "values" of all iteration parameters with smaller subindices, as they have been set earlier. Thus, we may assume that each iteration parameter is "arbitrarily larger" than all the parameters with smaller indices.

We now re-define the conjunctive and disjunctive components of formulae in $\mathrm{CTL}^{par}_{\mathbf{FB}}$ as in the table below (with no changes for the formulae that are not listed there). Note here that, since we give no semantics for the formulae $\mathbf{G}^{\mathbf{n}_i}_{\mathsf{Q}}(\varphi)$ and $\mathbf{U}^{\mathbf{n}_i}_{\mathsf{Q}}(\varphi,\psi)$, their components are "symbolical" and are only defined in order to be used with the procedure FullExpansion.

formulae	conjunctive component	formulae	disjunctive component
$\mathsf{AG}\,\varphi$	$\{\varphi, \mathsf{AX}\,\mathsf{AG}\,\varphi\}$	$\neg\mathsf{AG}\,\varphi$, $\neg\mathbf{G}^{\mathbf{n}_i}_{\mathsf{A}}(\varphi)$	$\{\neg\varphi, \neg\mathsf{AX}\,\mathbf{G}^{\mathbf{n}_i}_{\mathsf{A}}(\varphi)\}$
$\mathsf{EG}\,\varphi$, $\mathbf{G}^{\mathbf{n}_i}_{\mathsf{E}}(\varphi)$	$\{\varphi, \mathsf{EX}\,\mathbf{G}^{\mathbf{n}_i}_{\mathsf{E}}(\varphi)\}$	$\neg\mathsf{EG}\,\varphi$, $\neg\mathbf{G}^{\mathbf{n}_i}_{\mathsf{E}}(\varphi)$	$\{\neg\varphi, \neg\mathsf{EX}\,\mathbf{G}^{\mathbf{n}_i}_{\mathsf{E}}(\varphi)\}$
$\neg\mathsf{E}(\varphi\,\mathsf{U}\,\psi)$	$\{\neg\psi, \neg\varphi \vee \neg\mathsf{EX}\,\mathsf{E}(\varphi\,\mathsf{U}\,\psi)\}$	$\mathsf{E}(\varphi\,\mathsf{U}\,\psi)$, $\mathbf{U}^{\mathbf{n}_i}_{\mathsf{E}}(\varphi,\psi)$	$\{\psi, \varphi \wedge \mathsf{EX}\,\mathbf{U}^{\mathbf{n}_i}_{\mathsf{E}}(\varphi,\psi)\}$
$\neg\mathsf{A}(\varphi\,\mathsf{U}\,\psi)$,$\neg\mathbf{U}^{\mathbf{n}_i}_{\mathsf{A}}(\varphi,\psi)$	$\{\neg\psi, \neg\varphi \vee \neg\mathsf{AX}\,\mathbf{U}^{\mathbf{n}_i}_{\mathsf{A}}(\varphi,\psi)\}$	$\mathsf{A}(\varphi\,\mathsf{U}\,\psi)$, $\mathbf{U}^{\mathbf{n}_i}_{\mathsf{A}}(\varphi,\psi)$	$\{\psi, \varphi \wedge \mathsf{AX}\,\mathbf{U}^{\mathbf{n}_i}_{\mathsf{A}}(\varphi,\psi)\}$

The cases with two formulae (e.g. $\mathsf{EG}\,\varphi$ and $\mathbf{G}^{\mathbf{n}_i}_{\mathsf{E}}(\varphi)$), in the table above, have the same component for both formulae. However, the *index* i chosen for the iteration parameter $\mathbf{n}_i$ with the second component (e.g. $\mathsf{EX}\,\mathbf{G}^{\mathbf{n}_i}_{\mathsf{E}}(\varphi)$) is chosen differently for the formula on the left (e.g. $\mathsf{EG}\,\varphi$) and the formula on the right (e.g. $\mathbf{G}^{\mathbf{n}_i}_{\mathsf{E}}(\varphi)$); see below.

- (Left) Here we need to introduce a *new* parameter $\mathbf{n}_i$ for the component. The value of the index $i \in \mathbb{N}^+$ depends on the context; see the construction of the set $\mathrm{FE}(\Delta)$ below.
- (Right) Here the *same* iteration parameter $\mathbf{n}_i$ (with the same index i) that occurs in the original formula is used for the second component as well.

The notions of extended closure and full expansions for any set of formulae Δ in $\mathrm{CTL}^{par}_{\mathbf{FB}}$ are defined as for $\mathrm{CTL}_{\mathbf{FB}}$. When creating the set $\mathrm{FE}(\Delta)$ we may need to introduce new iteration parameters $\mathbf{n}_i$ for the components of formulae in Δ (e.g. in the case of $\mathsf{EG}\,\varphi$ in the table above). When this happens, the index $i \in \mathbb{N}^+$ of $\mathbf{n}_i$ is chosen to be *the smallest integer i that is greater than* all other indexes of iteration parameters that currently occur in Δ. Note that the same index i can be used for all formulae for which we need to introduce a new parameter $\mathbf{n}_i$ (at the same time). Therefore in $\mathrm{FE}(\Delta)$ there is at most one parameter $\mathbf{n}_i$ that does not occur in Δ. Also note that for a finite set Δ, the set $\mathrm{FE}(\Delta)$ is also finite.

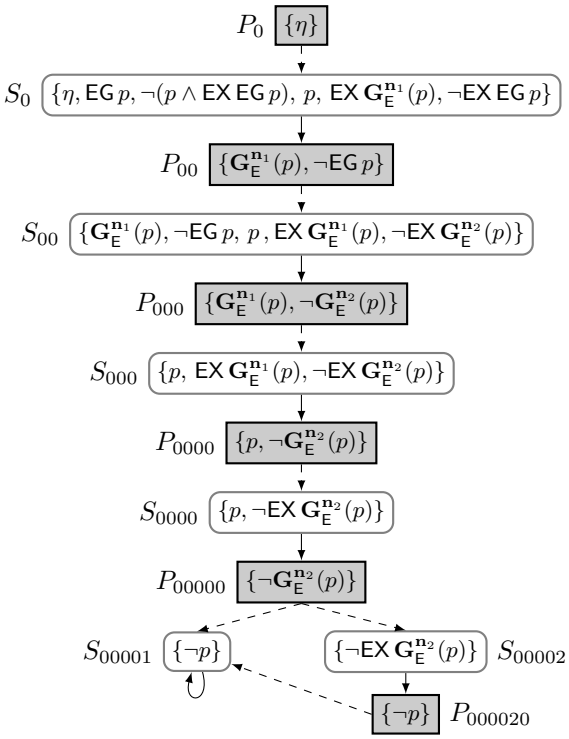

Figure 5 Finitary pretableau for $\eta = \mathsf{EG}\,p \wedge \neg(p \wedge \mathsf{EX}\,\mathsf{EG}\,p)$.

Finally we consider the case when a successor prestate Γ would be created for a state Δ but there already exists a prestate Γ' with the same label as Γ. If no iteration parameter $\mathbf{n}_i$ occurs in Γ, then we proceed as normally by adding an arrow $\Delta \to \Gamma'$. Else, we apply the following "parameter elimination" procedure:

ParElim: Let i be the *smallest* integer for which $\mathbf{n}_i$ occurs (possibly many times) in Γ. Now we first create a set $\Gamma_{/i}$ by replacing all formulae in Γ that contain $\mathbf{n}_i$ with the formulae in the table below. Then we add a successor prestate of Δ with the label $\Gamma_{/i}$ and then continue the finitary pretableau construction as normally.

formula	replacement	formula	replacement
$\mathbf{G}^{\mathbf{n}_i}_{\mathsf{E}}(\varphi)$	φ	$\neg\mathbf{G}^{\mathbf{n}_i}_{\mathsf{A}}(\varphi)$, $\neg\mathbf{G}^{\mathbf{n}_i}_{\mathsf{E}}(\varphi)$	$\neg\varphi$
$\mathbf{U}^{\mathbf{n}_i}_{\mathsf{E}}(\varphi,\psi)$, $\mathbf{U}^{\mathbf{n}_i}_{\mathsf{A}}(\varphi,\psi)$	ψ	$\neg\mathbf{U}^{\mathbf{n}_i}_{\mathsf{A}}(\varphi,\psi)$	$\neg\psi$

Note that here $\mathbf{n}_i$ is intuitively "lowered to zero", as e.g. $\neg\mathbf{G}^0_{\mathsf{A}}(\varphi) = \neg\mathbf{G}^0_{\mathsf{E}}(\varphi) = \neg\varphi$.

Apart from the changes described above, the finitary pretableau for $\mathrm{CTL}^{par}_{\mathbf{FB}}$ is constructed in the same way as the infinitary pretableau for $\mathrm{CTL}_{\mathbf{FB}}$.

▶ **Example 27.** The formula $\eta = \mathsf{EG}\,p \wedge \neg(p \wedge \mathsf{EX}\,\mathsf{EG}\,p)$ has the finitary pretableau given in Figure 5. Here the procedure ParElim is applied when the prestates P_{0000} and P_{000020} are created. Note that this pretableau is finite and rather simple, but it still (in some sense) encodes all the relevant information of the corresponding infinitary pretableau in Example 17.

The prestate and state eliminations are done exactly as in the infinitary tableau for $\mathrm{CTL}_{\mathbf{FB}}$; they always terminate in finite number of steps and produce a finite final tableau. We will show that the final finitary tableau for a formula φ is open if and only if φ is satisfiable.

4.2 Equivalence of the two tableaux and their complexity

▶ **Theorem 28.** *The infinitary tableau for a formula $\eta \in \mathrm{CTL}_{\boldsymbol{FB}}$ is open if and only if the finitary tableau for η is open.*

A detailed proof sketch for this equivalence is given in the appendix, but we explain here the main ideas. The infinite branchings in the infinitary tableau are replaced with only a single branch in the finitary tableau. For example, for $\neg\mathsf{EG}\,\varphi$, (typically) in the infinitary tableau there is an infinite branching such that for each $n \in \mathbb{N}$, there is an offspring state in which $\neg\mathsf{EX}\,\mathbf{G}^n_{\mathsf{E}}(\varphi)$ occurs, while in the finitary tableau, the single $\mathrm{CTL}^{par}_{\mathbf{FB}}$-formula $\neg\mathsf{EX}\,\mathbf{G}^{\mathbf{n}_i}_{\mathsf{E}}(\varphi)$, for some $i \in \mathbb{N}^+$, is used in the place of all these formulae. We can show that for sufficiently large values of n, all the corresponding branches in the infinitary tableau have essentially the same structure. Furthermore, it suffices to only consider these high values of n when checking whether a state gets eliminated from the infinitary tableau.

Note that in the rules of the finitary tableau, the iteration parameter $\mathbf{n}_i$ is treated as an (actual) iteration counter n except that its value is not lowered normally in transitions. But if the same state is to be repeated in the finitary tableau, then one of the iteration parameters is replaced with the value zero (by the procedure $\mathsf{ParElim}$). In the corresponding situation in the infinitary tableau, the similar states could be repeated until some of the iteration counters goes to zero. When considering only sufficiently large values of n, we may suppose that n is larger than all the iteration counters that have been set before the value of n has been fixed. Therefore n goes to zero only after all iteration counters set earlier have gone to zero. The same thing happens with the finitary tableau rules, as the iteration parameters $\mathbf{n}_i$ are lowered to zero in the order in which they have been introduced.

As seen above, the iteration parameter $\mathbf{n}_i$ is treated as if it had "as large a value as possible." Therefore we can show that the branches in the finitary tableau where $\mathbf{n}_i$ is used in the place of n, have essentially the same structure as the branches in the infinitary tableau where sufficiently large values of n are used. As only these branches are relevant for checking whether the infinitary tableau closes, it follows that the two tableaux are equivalent.

▶ **Corollary 29.** *The finitary tableau for $\mathrm{CTL}_{\boldsymbol{FB}}$ is sound and complete.*

▶ **Theorem 30.** *The complexity of the finitary tableau for $\mathrm{CTL}_{\boldsymbol{FB}}$ is* EXPTIME*-complete.*

The upper bound is easy to see by first noticing that the set of different labels of states in the finitary tableau is exponential with respect to the size of the input formula. The lower bound is obtained by a simple translation from the bi-modal logic with an additional reflexive-transitive closure operator. See more details in the Appendix.

▶ **Corollary 31.** *The satisfiability problem of $\mathrm{CTL}_{\boldsymbol{FB}}$ is decidable and* EXPTIME*-complete.*

5 Concluding remarks

The motivation for the logic $\mathrm{CTL}_{\mathbf{FB}}$ introduced here was two-fold: natural game-theoretic semantics and uniform boundedness of the time limit for satisfaction of eventualities across all branches. Both apply well beyond CTL, e.g. also to produce respective versions of the logics CTL* and ATL, to be studied further. While $\mathrm{CTL}_{\mathbf{FB}}$ is a simple semantic variation of CTL, it turns out to display some essentially different semantic features, the most striking being the lack of finite model property. That, in particular, makes capturing its validities more difficult and requiring an infinitary Hilbert-style axiomatization which will be presented in a subsequent work. An implementation of the finitary tableau method developed here will

also be considered in future. Lastly, symbolic model checking of $\mathrm{CTL}_{\mathbf{FB}}$ on some classes of finitely presented infinite models is another natural direction for future work.

Acknowledgements We thank the reviewers for useful remarks.

References

1. R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002.
2. Armin Biere, Alessandro Cimatti, Edmund M. Clarke, and Yunshan Zhu. Symbolic model checking without BDDs. In *Proc. of TACAS'99*, pages 193–207, 1999.
3. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. CUP, 2001.
4. E. M. Clarke and E. A. Emerson. Design and synthesis of synchronisation skeletons using branching time temporal logic. In *Logics of Programs*, pages 52–71. Springer, 1981.
5. Stéphane Demri, Valentin Goranko, and Martin Lange. *Temporal Logics in Computer Science*. Cambridge University Press, 2016.
6. Valentin Goranko, Antti Kuusisto, and Raine Rönnholm. Game-theoretic semantics for alternating-time temporal logic. In *Proc. of AAMAS 2016*, pages 671–679, 2016.
7. Valentin Goranko, Antti Kuusisto, and Raine Rönnholm. Game-theoretic semantics for alternating-time temporal logic. arXiv:1602.07667 [math.LO], submitted, 2017.
8. Wojciech Penczek, Bozena Wozna, and Andrzej Zbrzezny. Bounded model checking for the universal fragment of CTL. *Fundam. Inform.*, 51(1-2):135–156, 2002.
9. Bozena Wozna. ACTLS properties and bounded model checking. *Fundam. Inform.*, 63(1):65–87, 2004.
10. Wenhui Zhang. Bounded semantics. *Theor. Comput. Sci.*, 564:1–29, 2015.

A Appendix

Proof sketch of Theorem 16

One direction is immediate by Lemma 15, for $\Gamma = ecl(\eta)$. For the converse, suppose $\eta \in \mathrm{H}(s_0)$ for some state s_0 in some Hintikka structure $\mathcal{H} = (\mathrm{S}, \mathrm{R}, \mathrm{H})$. We define the interpreted transition system $\mathcal{M} = (\mathrm{S}, \mathrm{R}, \Phi, \ell)$ where ℓ is a state description in S defined as follows: $\ell(s) := \mathrm{PROP} \cap \mathrm{H}(s)$.

We show by induction on the main components of $\varphi \in ecl(\eta)$ that for every $s \in \mathrm{S}$:

if $\varphi \in \mathrm{H}(s)$ then $\mathcal{M}, s \models \varphi$.

The details are fairly routine. The cases of the temporal operators follow directly from Lemma 4.

Now, since $\eta \in \mathrm{H}(s_0)$ we have that $\mathcal{M}, s_0 \models \eta$.

Proof sketch of Lemma 21

Suppose the elimination phase terminates at some ordinal β, i.e. no states are eliminated from $\mathcal{T}^\eta_\beta$. It suffices to show, by transfinite induction on the ordinal number of elimination steps $\alpha \leq \beta$ taken in the elimination phase, that no satisfiable state $\Delta \in \mathcal{T}^\eta_\alpha$ is removed by an application of $\textsc{StateElim}^{\mathrm{CTL}_{\mathbf{FB}}}$ to $\mathcal{T}^\eta_\alpha$. For that purpose we will prove a somewhat stronger inductive claim, viz., that for every ordinal $\alpha \leq \beta$:

1. If $\Delta \in \mathcal{T}^\eta_\alpha$ is satisfiable, then for any $\mathsf{EX}\,\varphi \in \Delta$ there is a satisfiable state $\Psi_\varphi \in \mathcal{T}^\eta_\alpha$ such that $\Delta \xrightarrow{\mathsf{EX}\,\varphi} \Psi_\varphi$ in $\mathcal{T}^\eta_\alpha$.

2. If $\Delta \in \mathcal{T}_\alpha^\eta$ is satisfiable, then for any $\neg\mathsf{AX}\,\varphi \in \Delta$ there is a satisfiable state $\Psi_{\neg\varphi} \in \mathcal{T}_\alpha^\eta$ such that $\Delta \xrightarrow{\neg\mathsf{AX}\,\varphi} \Psi_{\neg\varphi}$ in $\mathcal{T}_\alpha^\eta$.
3. All satisfiable states in $\mathcal{T}_0^\eta$ are still present in $\mathcal{T}_\alpha^\eta$.

Note that the inductive hypothesis refers simultaneously to all satisfiable $\Delta \in \mathcal{T}_\alpha^\eta$ and all successor formulae $\mathsf{EX}\,\varphi \in \Delta$ and $\neg\mathsf{AX}\,\varphi \in \Delta$. We will only give the proof of claim 1; claim 2 is completely analogous, and then claim 3 follows immediately from these.

Assume the claim holds for all $\gamma < \alpha$ and consider 3 cases for α.

1. Let $\alpha = 0$. Take a satisfiable $\Delta \in \mathcal{T}_\alpha^\eta$ and $\mathsf{EX}\,\varphi \in \Delta$, Then, $\mathcal{M}, s \models \Delta$ for some rooted ITS $(\mathcal{M}, s)$. Recall, that all states $\Delta' \in \mathcal{T}_0^\eta$ such that $\Delta \xrightarrow{\mathsf{EX}\,\varphi} \Delta'$ in $\mathcal{T}_0^\eta$ are obtained as full expansions of the prestate $\Gamma = \{\varphi\} \cup \{\,\psi \mid \mathsf{AX}\,\psi \in \Delta\,\} \cup \{\,\neg\psi \mid \neg\mathsf{EX}\,\psi \in \Delta\,\}$. Since $\mathsf{EX}\,\varphi \in \Delta$, we have that $\mathcal{M}, s \models \mathsf{EX}\,\varphi$, hence there is an R-successor r of s in $\mathcal{M}$ such that $\mathcal{M}, r \models \varphi$. By the truth definition for successor formulae, it follows that $\mathcal{M}, r \models \Gamma$ because $\{\,\psi \mid \mathsf{AX}\,\psi \in \Delta\,\} \cup \{\,\neg\psi \mid \neg\mathsf{EX}\,\psi \in \Delta\,\}$ is satisfied at every R-successor of s in $\mathcal{M}$. Then, by Lemma 20, at least one full expansion Ψ_φ of Γ is satisfied by $(\mathcal{M}, r)$, and, by construction of the initial tableau, there is a state (with label) Ψ_φ in $\mathcal{T}_0^\eta$ such that $\Delta \xrightarrow{\mathsf{EX}\,\varphi} \Psi_\varphi$. Therefore, the state Δ cannot be removed from $\mathcal{T}_0^\eta$ by an application of the rule $\textsc{StateElim}^{\mathrm{CTL}_{\mathbf{FB}}}$.
2. Let α be a successor ordinal. Assuming the claim holds for all $\gamma < \alpha$, take a satisfiable $\Delta \in \mathcal{T}_\alpha^\eta$. For any $\mathsf{EX}\,\varphi \in \Delta$, by the argument for $\alpha = 0$ there is a satisfiable state Ψ_φ in $\mathcal{T}_0^\eta$ such that $\Delta \xrightarrow{\mathsf{EX}\,\varphi} \Psi_\varphi$ in $\mathcal{T}_0^\eta$, and hence, by the inductive hypothesis, Ψ_φ has remained intact in $\mathcal{T}_\alpha^\eta$. Therefore, Δ cannot be removed from $\mathcal{T}_\alpha^\eta$ by an application of the rule $\textsc{StateElim}^{\mathrm{CTL}_{\mathbf{FB}}}$.
3. In the case when α is a limit ordinal, $\mathcal{T}_\alpha^\eta$ is the intersection of all $\mathcal{T}_\gamma^\eta$ for $\gamma < \alpha$ and the argument is essentially the same as in the previous case, but now applied to each $\mathcal{T}_\gamma^\eta$ for $\gamma < \alpha$.

Proof of Lemma 20. Follows easily from Proposition 11.

Proof of Theorem 22. Follows immediately from Lemmas 20 and 21.

Proof sketch of Theorem 23

It suffices to show how from the open final tableau $\mathcal{T}^\eta$ one can construct a Hintikka structure satisfying η. That construction for $\mathrm{CTL}_{\mathbf{FB}}$ is similar to the construction for CTL described in [5, Chapter 13], to which the reader is referred for further details. It consists in extracting a subgraph of the final tableau $\mathcal{T}^\eta$ which contains an initial state with the input formula and represents any 'logical branch' of the search procedure encoded by $\mathcal{T}^\eta$. Such a 'logical branch' can be extracted by selecting just one alternative state from every set of states in $\mathcal{T}^\eta$ generated in the pretableau construction phase as full expansions of the same prestate. The resulting subgraph itself can be taken as a Hintikka structure, with a labelling function assigning to each state its own label. The proof that it is a Hintikka structure for $ecl(\eta)$ is straightforward by the construction of the tableau.

Proof sketch of Proposition 26

Since there are only countably many formulae, each state in a tableau is a finite or countably infinite set of formulae. Thus each state has at most countably many EX-formulae that require a successor prestate, one prestate for each EX-formula. In the process of selecting

'logical branches' (cf. the proof of Theorem 23), we eliminate all except for one of the states generated from a single prestate by the full expansion procedure. Thus, we observe that the out-degree of each state in the ultimate model is at most countably infinite and each such state is constructed after finitely many steps from the initial state. Therefore, the whole model is a countable union of countable sets of states.

Proof sketch of Theorem 28

For simplicity we suppose here that η does not have subformulae of the form $\mathsf{E}(\varphi \,\mathsf{U}\, \psi)$ or $\mathsf{A}(\varphi \,\mathsf{U}\, \psi)$, since these can be treated in a very similar way to the formulae $\mathsf{EG}\,\varphi$ and $\mathsf{AG}\,\varphi$. As the rules of the infinitary and the finitary tableau differ now only in those rules related to $\mathsf{EG}\,\varphi$, $\neg\mathsf{AG}\,\varphi$ and $\neg\mathsf{EG}\,\varphi$, we compare how these formulae are treated in the two tableaux. In the infinitary tableau, the rules for these formulae typically create infinite branchings, with a branch for every iteration counter $n \in \mathbb{N}$, while in the finitary tableau an iteration parameter $\mathbf{n}_i$ is used in the place of every n. We aim to show that it suffices to consider only certain 'sufficiently large' values of n in order to determine whether the infinitary tableau closes or not. Furthermore, we will argue that a branch with such a large value of n has (essentially) the same structure as a branch in the finitary tableau with the iteration parameter $\mathbf{n}_i$. The equivalence of the two tableaux will then be easy to see from these observations.

We will first define explicitly what we mean above by 'sufficiently large' value of n. For this, consider a prestate Γ whose label contains some formulae of the form $\mathsf{EG}\,\varphi$, $\neg\mathsf{AG}\,\varphi$ or $\neg\mathsf{EG}\,\varphi$. Let $m \in \mathbb{N}$ be the *smallest integer that is greater than all iteration counters* which currently occur in the label of Γ (in the infinitary tableau). We now define $n := m + k$, where k is the size of the formula η (i.e. the number of symbols in η). We will show that this (or any greater) value of n is the only one that needs to be considered as iteration counter when we create offspring states of Γ.

We first consider the case of the formula $\neg\mathsf{EG}\,\varphi$. If $\neg\mathsf{EG}\,\varphi$ occurs in some prestate Γ of the infinitary tableau, then for each value of $n \in \mathbb{N}$ the formula $\neg\mathsf{EX}\,\mathbf{G}_{\mathsf{E}}^{n-1}(\varphi)$ occurs in some offspring state of Γ; excluding those values of n that create patently inconsistent sets. The only difference between the different values of n here is that n is lowered to zero at a different stage in the tableau construction process. We first observe that if a state with certain value of n survives the state elimination process, then all the larger values will survive as well. Therefore if a state with some lower than $m + k$ value is not eliminated, then $m + k$ is not eliminated either. But on the other hand, if a state with some larger than $m + k$ value of n is not eliminated, then the state with $n = m + k$ cannot be eliminated either. In order to see this, we first observe the following facts:

1. $n = m + k$ cannot go down to zero before all iteration counters that have been set before n have gone down to zero.
2. After all other other iteration counters set before $n = m + k$ have gone to zero, all the nested temporal operators EX or AX will be removed before n goes to zero.

So after $n = m + k$ has gone to zero, the elimination of the reached state depends only on the *values set after* n. If this state will be eliminated, then it is easy to see that it would be eliminated even if some larger than $m + k$ value of n was selected, since all the values, set after fixing n, can always be arbitrarily larger than n.

The iteration parameter $\mathbf{n}_i$ in the finitary tableau is treated essentially the same way as the value $n = m + k$ in the infinitary tableau. This is simply because $\mathbf{n}_i$ cannot be 'lowered to zero' unless a state with the same label is to be repeated in the tableau, and all the iteration parameters set before $\mathbf{n}_i$ are first eliminated (recall the procedure $\mathsf{ParElim}$). Hence, it is easy

to see that a state with $n = m + k$ in the infinitary tableau is eliminated if and only if the state with $\mathbf{n}_i$ in the finitary tableau is eliminated.

We then consider the case of $\neg\mathsf{AG}\,\varphi$. If $\neg\mathsf{AG}\,\varphi$ occurs in some prestate Γ of the infinitary tableau, then for each value of $n \in \mathbb{N}$ the formula $\neg\mathsf{AX}\,\mathbf{G}^n_{\mathsf{A}}(\varphi)$ occurs in some offspring state of Γ (except those values of n that create patently inconsistent sets). Hereafter the reasoning can be done similarly as for the formula $\neg\mathsf{EG}\,\varphi$.

Finally we consider the case of $\mathsf{EG}\,\varphi$. If $\mathsf{EG}\,\varphi$ occurs in some prestate Γ of the infinitary tableau, then all the formulae $\mathsf{EX}\,\mathbf{G}^{n-1}_{\mathsf{E}}(\varphi)$ $(n \in \mathbb{N})$ occur in *every* offspring state of Γ. Consider some offspring state Δ of Γ. When creating successor prestates for Δ, we must create a successor prestate Γ_n for each value $n \in \mathbb{N}$ and include $\mathbf{G}^{n-1}_{\mathsf{E}}(\varphi)$ in the label Γ_n. In order for Δ to survive the state elimination process, the following must hold for *every* $n \in \mathbb{N}$:

Γ_n has an offspring state Δ_n which survives the state elimination process. (C_n)

We first observe that if the condition (C_n) does not hold for some $n < m + k$, then it cannot hold for $n = m + k$, either. But, with similar observations as in the case for $\neg\mathsf{EG}\,\varphi$, we can also see that if (C_n) does not hold for some $n > m + k$, then it cannot hold for $n = m + k$ either. Hence it suffices that we only consider the case $n = m + k$ to see whether (C_n) holds for every $n \in \mathbb{N}$. With similar reasoning as above we see that this case is essentially the same as the use of iteration parameter $\mathbf{n}_i$ in the finitary tableau.

Proof sketch of Theorem 30

(A) Upper bound. The number of formulae (possibly containing iteration parameters) that are used in a tableau for an input formula η, is linear in $|\eta|$. Thus, the number of different labels of states (prestates) is exponential in $|\eta|$. Therefore, the number of nodes appearing in the tableau construction is, likewise, exponential in $|\eta|$. The different construction maneuvers are computationally straightforward and can easily be done deterministically. Therefore, only exponentially many (in $|\eta|$) steps are required for carrying out the tableau procedure.

(B) Lower bound. Consider standard modal logic with the reflexive and transitive closure operator, i.e., the logic generated by the grammar

$$\varphi ::= p \,|\, \neg\varphi \,|\, (\varphi \wedge \varphi) \,|\Diamond\varphi \,|\, \Diamond^*\varphi$$

where p is a proposition symbol. The semantics is the standard one, with $\Diamond^*$ denoting the diamond interpreted by the reflexive-transitive closure of the binary relation interpreting $\Diamond$. It is well known that the satisfiability problem of this logic is EXPTIME-complete; see, e.g., [3]. Consider the simple translation T from this logic into $\mathrm{CTL}_{\mathbf{FB}}$, preserving atomic propositions and Boolean connectives, and mapping $T(\Diamond\varphi)$ to $\mathsf{EX}\,T(\varphi)$ and $T(\Diamond^*\varphi)$ to $\mathsf{EF}\,T(\varphi)$. It easy to see that this translation preserves equivalence, i.e., precisely the same points of any model satisfy φ and $T(\varphi)$.