Optimal Tableaux-based Decision Procedure for Testing Satisfiability in the Alternating-time Temporal Logic ATL$^+$

Serenella Cerrito[1], Amélie David[1], and Valentin Goranko[2]

Address
[1]Laboratoire IBISC - Université Evry Val-d'Essonne, France
{serena.cerrito,adavid}@ibisc.univ-evry.fr
[2]Department of Applied Mathematics and Computer Science
Technical University of Denmark,
[2]University of Johannesburg, South Africa

Abstract. We develop a sound, complete and practically implementable tableaux-based decision method for constructive satisfiability testing and model synthesis in the fragment ATL$^+$ of the full Alternating time temporal logic ATL*. The method extends in an essential way a previously developed tableaux-based decision method for ATL and works in 2EXPTIME, which is the optimal worst case complexity of the satisfiability problem for ATL$^+$. We also discuss how suitable parameterizations and syntactic restrictions on the class of input ATL$^+$ formulae can reduce the complexity of the satisfiability problem.

1 Introduction

The Alternating-time temporal logic ATL* was introduced and studied in [1] as a multi-agent extension of the branching time temporal logic CTL*, where the path quantifiers are generalized to "strategic quantifiers", indexed with coalitions of agents A and ranging over all computations enabled by a given collective strategy of A. ATL* was proposed as logical framework for specification and verification of properties of open systems modeled as concurrent game models, in which all agents effect state transitions collectively, by taking simultaneous actions at each state. The language of ATL* allows expressing statements of the type *"Coalition A has a collective strategy to guarantee the satisfaction of the objective Φ on every play enabled by that strategy"*. The syntactic fragment ATL of ATL* allows only state formulae, where all occurrences of temporal operators must be immediately preceded by strategic quantifiers. The fragment ATL$^+$ of ATL* extends ATL by allowing any Boolean combinations of ATL objectives in the scope of a strategic quantifier. It is considerably more expressive than ATL, which is reflected in the high – 2EXPTIME – worst case complexity lower bound of the satisfiability problem for ATL$^+$ (inherited from the lower bound for CTL$^+$, see [8]) as opposed to the EXPTIME-completeness of the satisfiability problem for ATL [4, 11]. The matching 2EXPTIME upper bound is provided by the automata-based method for deciding satisfiability in the full ATL*, developed in [10].

The contribution of this paper is the development of a sound, complete and terminating tableaux-based decision method for constructive satisfiability testing of ATL^+ formulae, which we also claim to be intuitive, conceptually simple and transparent, as well as practically implementable and even manually usable, despite the inherently high worst-case complexity of the problem. The tableaux method presented here is based on the general methodology going back to [9] and [12]. It was further developed for ATL in [7] to which the reader is referred for more details, and a recent implementation is reported in [3]. The tableaux method for ATL^+ is an essential extension of the one for ATL, as it has to deal with much more complex (and computationally expensive) path objectives that can be assigned to the agents. It is also rather different from the above mentioned automata-based method in [10].

The paper is structured as follows. In Section 2 we offer brief technical preliminaries on concurrent game models, syntax and semantics of ATL^* and ATL^+. Section 3 develops the technical machinery needed for the presentation of the tableaux method itself in Section 4. Section 5 contains the main results related to termination, soundness, completeness and complexity of the procedure. In Section 6 we offer a brief comparison with the automata-based method in [10].

For lack of space, we only provide here very brief sketches of the proofs of the soundness, completeness and some other technical claims. A full version of this paper, including detailed proofs, is available as a technical report [2].

2 Preliminaries

We assume that the reader has basic familiarity with the branching time logic CTL^*, see e.g. [5]. Also, basic knowledge on ATL^* [1] and the tableaux-based decision procedure for ATL in [7], on which this paper builds, would be beneficial.

2.1 Concurrent game models, strategies and co-strategies

A **concurrent game model** [1] (CGM) is a tuple
$\mathcal{M} = (\mathbb{A}, \mathsf{St}, \{\mathsf{Act_a}\}_{\mathsf{a}\in\mathbb{A}}, \{\mathsf{act_a}\}_{\mathsf{a}\in\mathbb{A}}, \mathsf{out}, \mathsf{Prop}, \mathsf{L})$ comprising:

- a finite, non-empty set of *players (agents)* $\mathbb{A} = \{1, \dots, k\}$
- a non-empty set of *states* St,
- a set of actions $\mathsf{Act_a} \neq \emptyset$ for each $\mathsf{a} \in \mathbb{A}$.
  For any $A \subseteq \mathbb{A}$ we denote $\mathsf{Act}_A := \prod_{\mathsf{a}\in A} \mathsf{Act_a}$ and use σ_A to denote a tuple from Act_A. In particular, $\mathsf{Act}_\mathbb{A}$ is the set of all possible *action profiles* in $\mathcal{M}$.
- for each $\mathsf{a} \in \mathbb{A}$, a map $\mathsf{act_a} : \mathsf{St} \to \mathcal{P}(\mathsf{Act_a}) \setminus \{\emptyset\}$ defining for each state s the actions available to a at s,
- a partial *transition function* $\mathsf{out} : \mathsf{St} \times \mathsf{Act}_\mathbb{A} - \to \mathsf{St}$ that assigns deterministically a *successor (outcome) state* $\mathsf{out}(s, \sigma_\mathbb{A})$ to every state s and action profile $\sigma_\mathbb{A} = \langle \sigma_1, \dots, \sigma_\mathsf{k} \rangle$, such that $\sigma_\mathsf{a} \in \mathsf{act_a}(s)$ for every $\mathsf{a} \in \mathbb{A}$,
- a set of *atomic propositions* Prop, and a *labelling function* $\mathsf{L} : \mathsf{St} \to \mathcal{P}(\mathsf{Prop})$.

Concurrent game models represent multi-agent transition systems that function as follows: at any moment the system is in a given state, where each agent selects an action from those available to him at that state. All agents execute their actions synchronously and the combination of these actions together with the current state determine a transition to a unique successor state in the model. A *play* in a CGM is an infinite sequence of subsequent successor states, i.e., an infinite sequence $s_0 s_1 ... \in \mathsf{St}^\omega$ of states such that for each $i \geq 0$ there exists an action profile $\sigma_{\mathbb{A}} = \langle \sigma_1, \ldots, \sigma_{\mathsf{k}} \rangle$ such that $\mathsf{out}(s_i, \sigma_{\mathbb{A}}) = s_{i+1}$. A *history* is a finite prefix of a play. We denote by $\mathsf{Plays}_{\mathcal{M}}$ and $\mathsf{Hist}_{\mathcal{M}}$ respectively the set of plays and set of histories in $\mathcal{M}$. For a state $s \in \mathsf{St}$ we define $\mathsf{Plays}_{\mathcal{M}}(s)$ and $\mathsf{Hist}_{\mathcal{M}}(s)$ as the set of plays and set of histories with initial state s. Given a sequence of states λ, we denote by λ_0 its initial state, by λ_i its $(i+1)$th state, by $\lambda_{\leq i}$ the prefix $\lambda_0 ... \lambda_i$ of λ and by $\lambda_{\geq i}$ the suffix $\lambda_i \lambda_{i+1} ...$ of λ. When $\lambda = \lambda_0 ... \lambda_\ell$ is finite, we say that it has length ℓ and write $|\lambda| = \ell$. Further, we put $\mathrm{last}(\lambda) = \lambda_\ell$.

A *(perfect recall) strategy* for an agent a in $\mathcal{M}$ is a mapping $F_{\mathsf{a}} : \mathsf{Hist}_{\mathcal{M}} \to \mathsf{Act}_{\mathsf{a}}$ such that for all $h \in \mathsf{Hist}_{\mathcal{M}}$ we have $F_{\mathsf{a}}(h) \in \mathsf{act}_{\mathsf{a}}(\mathrm{last}(h))$. Intuitively, it assigns an admissible action for agent a after any history h of the game. We denote by $\mathsf{Strat}_{\mathcal{M}}(\mathsf{a})$ the set of strategies of agent a. A (collective) strategy of a set (*coalition*) of agents $A \subseteq \mathbb{A}$ is a tuple $(F_{\mathsf{a}})_{\mathsf{a} \in A}$ of strategies, one for each agent in A. When $A = \mathbb{A}$ this is called a *strategy profile*. We denote by $\mathsf{Strat}_{\mathcal{M}}(A)$ the set of collective strategies of coalition A. A play $\lambda \in \mathsf{Plays}_{\mathcal{M}}$ is *consistent with a collective strategy* $F_A \in \mathsf{Strat}_{\mathcal{M}}(A)$ if for every $i \geq 0$ there exists an action profile $\sigma_{\mathbb{A}} = \langle \sigma_1, \ldots, \sigma_{\mathsf{k}} \rangle$ such that $\mathsf{out}(\lambda_i, \sigma_{\mathbb{A}}) = \lambda_{i+1}$ and $\sigma_{\mathsf{a}} = F_{\mathsf{a}}(\lambda_{\leq i})$ for all $\mathsf{a} \in A$. The set of plays with initial state s that are consistent with F_A is denoted $\mathsf{Plays}_{\mathcal{M}}(s, F_A)$. For any coalition $A \subseteq \mathbb{A}$ and a given CGM $\mathcal{M}$ and state $s \in \mathsf{St}$, an *A-co-move* at s in $\mathcal{M}$ is a mapping $\mathsf{Act}^c_A : \mathsf{Act}_A \to \mathsf{Act}_{\mathbb{A} \setminus A}$ that assigns to every collective action of A at the state s a collective action at s for the complementary coalition $\mathbb{A} \setminus A$. Likewise, an *A-co-strategy* in $\mathcal{M}$ is a mapping $F^c_A : \mathsf{Strat}_{\mathcal{M}}(A) \times \mathsf{St} \to \mathsf{Act}_{\mathbb{A} \setminus A}$ that assigns to every collective strategy of A and a state $s \in \mathsf{St}$ a collective action at s for $\mathbb{A} \setminus A$.

2.2 The logic ATL* and fragments

The logic ATL^* is a multi-agent extension of CTL^* with *strategic quantifiers* $\langle\!\langle A \rangle\!\rangle$ indexed with coalitions A of agents. There are two types of formulae in ATL^*: *state formulae*, that are evaluated at states, and *path formulae*, that are evaluated on plays. To simplify the presentation we will work with formulae in negation normal form over a fixed set Prop of atomic propositions and primitive temporal operators *Always* $\Box$ and *Until* $\mathcal{U}$. The syntax of the full language ATL^* and its fragments ATL^+ and ATL can then be defined as follows, where $l \in \mathsf{Prop} \cup \{\neg p \mid p \in \mathsf{Prop}\}$ is a literal, $\mathbb{A}$ is a fixed set of agents and $A \subseteq \mathbb{A}$:

$$\textit{State formulae: } \varphi := l \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid \langle\!\langle A \rangle\!\rangle \Phi \mid [\![A]\!] \Phi \qquad (1)$$

$$\mathsf{ATL}^*\textit{-path formulae: } \Phi := \varphi \mid \bigcirc \Phi \mid \Box \Phi \mid (\Phi \mathcal{U} \Phi) \mid (\Phi \vee \Phi) \mid (\Phi \wedge \Phi) \qquad (2)$$

$$\mathsf{ATL}^+\textit{-path formulae: } \Phi := \varphi \mid \bigcirc \varphi \mid \Box \varphi \mid (\varphi \mathcal{U} \varphi) \mid (\Phi \vee \Phi) \mid (\Phi \wedge \Phi) \qquad (3)$$

$$\mathsf{ATL}\textit{-path formulae: } \Phi := \bigcirc \varphi \mid \Box \varphi \mid (\varphi \mathcal{U} \varphi) \qquad (4)$$

Note that the state formulae have the same definition but define different sets in all 3 cases. To keep the notation lighter, we will list the members of the set A in $\langle\langle A \rangle\rangle$ without using $\{\}$. When the length of a formula is measured, A will be assumed given by a bit vector. Parentheses will be omitted whenever safe, but they will be important when conjunctions and disjunctions are composed.

Hereafter, we use φ, ψ, η to denote arbitrary state formulae and Φ, Ψ to denote path formulae. By an ATL^+ formula we will mean by default a *state* formula of ATL^+; likewise for ATL. We define $\top := p \vee \neg p$, $\bot := \neg\top$ and the temporal operators *Sometime* $\Diamond$ by $\Diamond\varphi := \top\mathcal{U}\varphi$ and *Release* $\mathcal{R}$ by $\varphi\mathcal{R}\psi := \Box\varphi \vee \varphi\mathcal{U}(\varphi \wedge \psi)$. Note, that $\langle\langle A \rangle\rangle\varphi\mathcal{R}\psi$ and $[\![A]\!]\varphi\mathcal{R}\psi$ are ATL^+ state formulae.

CTL^* can be regarded as the fragment of ATL^* where $\langle\langle \emptyset \rangle\rangle$ represents the path quantifier $\forall$ and $\langle\langle \mathbb{A} \rangle\rangle$ represents $\exists$. The semantics of ATL^* (inherited by ATL^+) is defined in a given CGM $\mathcal{M}$, state $s \in \mathcal{M}$ and a path λ in $\mathcal{M}$ just like the semantics of CTL^*, with the added clauses for the strategic quantifiers:

- $\mathcal{M}, s \models \langle\langle A \rangle\rangle\Phi$ iff there exists an A-strategy F_A such that, for all computations λ consistent with F_A, $\mathcal{M}, \lambda \models \Phi$.
- $\mathcal{M}, s \models [\![A]\!]\Phi$ iff there exists an A-co-strategy F^c_A such that, for all computations λ consistent with F^c_A, $\mathcal{M}, \lambda \models \Phi$.

Valid, satisfiable and equivalent formulae in ATL^* are defined as usual. Here are some important equivalences in LTL [5] and in ATL^* [1, 6], used further:

- $\Box\Psi \equiv \Psi \wedge \bigcirc\Box\Psi$; $\Phi \,\mathsf{U}\, \Psi \equiv \Psi \vee (\Phi \wedge \bigcirc(\Phi \,\mathsf{U}\, \Psi))$;
- $\langle\langle C \rangle\rangle\Box\Psi \equiv \Psi \wedge \langle\langle C \rangle\rangle \bigcirc \langle\langle C \rangle\rangle\Box\Psi$; $\langle\langle C \rangle\rangle\Phi \,\mathsf{U}\, \Psi \equiv \Psi \vee (\Phi \wedge \langle\langle C \rangle\rangle \bigcirc \langle\langle C \rangle\rangle\Phi \,\mathsf{U}\, \Psi)$;
- $[\![C]\!]\Box\Psi \equiv \Psi \wedge [\![C]\!] \bigcirc [\![C]\!]\Box\Psi$; $[\![C]\!]\Phi \,\mathsf{U}\, \Psi \equiv \Psi \vee (\Phi \wedge [\![C]\!] \bigcirc [\![C]\!]\Phi \,\mathsf{U}\, \Psi)$;
- $[\![\mathbb{A}]\!] \bigcirc \varphi \equiv \neg\langle\langle \mathbb{A} \rangle\rangle \bigcirc \neg\varphi \equiv \langle\langle \emptyset \rangle\rangle \bigcirc \varphi$; $\langle\langle A \rangle\rangle\langle\langle B \rangle\rangle\Phi \equiv \langle\langle B \rangle\rangle\Phi$;
- For every state formula φ: $\langle\langle A \rangle\rangle(\varphi\wedge\Psi) \equiv \varphi\wedge\langle\langle A \rangle\rangle\Psi$, $\langle\langle A \rangle\rangle(\varphi\vee\Psi) \equiv \varphi\vee\langle\langle A \rangle\rangle\Psi$.

Remark 1. It is known [1] that, when restricted to ATL formulae, the semantics above (based on perfect-recall strategies) is equivalent to the semantics based on positional (or memoryless) strategies, where the prescribed actions only depend on the current state, not on the whole history. This is no longer the case for ATL^+. For example, the formula $\langle\langle 1 \rangle\rangle\Diamond(p\wedge\langle\langle 1 \rangle\rangle\Diamond q) \rightarrow \langle\langle 1 \rangle\rangle(\Diamond p\wedge\Diamond q)$ is valid in the semantics with perfect-recall strategies (which can be freely composed) but not in the semantics with positional strategies (which cannot be freely composed).

Hereafter, we assume that the semantics is based on perfect-recall strategies.

Here we deal with the *(constructive) satisfiability decision problem* for ATL^+:

Given a state formula φ in ATL^+, does there exist a CGM $\mathcal{M}$ and a state s in $\mathcal{M}$ such that $\mathcal{M}, s \models \varphi$? If so, construct such a satisfying pair $(\mathcal{M}, s)$.

Remark 2. There are two variants of this satisfiability problem: *tight*, where it is assumed that all agents in the model are mentioned in the formula, and *loose*, where additional agents, not mentioned in the formula, are allowed in the model. These variants are really different, but the latter one is immediately reducible to the former, by adding just one extra agent a to the language. Furthermore, this extra agent can be easily added superfluously to the formula, e.g., by adding a conjunct $\langle\langle \mathsf{a} \rangle\rangle \bigcirc \top$, so we hereafter only consider the tight satisfiability version. For further details and discussion on this issue, see e.g., [11, 7].

3 Decomposition and closure of ATL^+ formulae

We partition the set of ATL^+ formulae into *primitive* and *non-primitive* formulae. The primitive formulae are $\top, \bot$, the literals and all ATL^+ *successor formulae*, of the form $\langle\langle A\rangle\rangle \bigcirc \psi$ or $[\![A]\!] \bigcirc \psi$, each with *successor component* ψ. The non-primitive formulae are classified as α-, β- and γ-formulae. An α-formula in our syntax is a conjunction $\varphi \wedge \psi$ with (conjunctive) α*-components* φ and ψ; a β-formula is a disjunction $\varphi \vee \psi$ with (disjunctive) β*-components* φ and ψ. The rest of the non-primitive formulae are classified as γ-formulae. That is, a γ-formula is one of the form $[\![A]\!]\Phi$ or $\langle\langle A\rangle\rangle\Phi$, where Φ is an ATL^+ path formula whose main operator is not $\bigcirc$. We note that, unlike [7], here we do not treat $\langle\langle A\rangle\rangle\Box\varphi$ as an α-formula nor $\langle\langle A\rangle\rangle\varphi\mathcal{U}\psi$ as a β-formula; both are γ-formulae.

The α- and β-formulae will be decomposed in the tableau as usual, while the case of γ-formulae $\langle\langle A\rangle\rangle\Phi$ and $[\![A]\!]\Phi$ is special and needs extra work, because their tableau decomposition will depend on the structure of Φ.

3.1 γ-decomposition and γ-components of γ-formulae

We denote the set of ATL^+ state formulae by $\mathsf{ATL}^+_\mathsf{s}$ and the set of ATL^+ path formulae by $\mathsf{ATL}^+_\mathsf{p}$. We will define a γ*-decomposition function* $\mathsf{dec} : \mathsf{ATL}^+_\mathsf{p} \to \mathcal{P}(\mathsf{ATL}^+_\mathsf{s} \times \mathsf{ATL}^+_\mathsf{p})$ with the following intuitive meaning: for any $\Phi \in \mathsf{ATL}^+_\mathsf{p}$ and pair $\langle\psi, \Psi\rangle \in \mathsf{dec}(\Phi)$, ψ is a state formula true at the current state and Ψ is a path formula expressing what must be true at the next state of a possible play starting at the current state. Thus, the set $\mathsf{dec}(\Phi)$ is interpreted as a disjunction describing all possible 'types of paths' starting from the current state and satisfying Φ. The definition of dec is recursive on ATL^+ path formulae, as follows.

⋆ $\mathsf{dec}(\varphi) = \{\langle\varphi, \top\rangle\}$, $\mathsf{dec}(\bigcirc\varphi) = \{\langle\top, \varphi\rangle\}$ for any ATL^+ state formula φ. The other base cases derive from the well-known LTL equivalences listed in 2.2:

⋆ $\mathsf{dec}(\Box\varphi) = \{\langle\varphi, \Box\varphi\rangle\}$ and $\mathsf{dec}(\varphi\mathcal{U}\psi) = \{\langle\varphi, \varphi\mathcal{U}\psi\rangle, \langle\psi, \top\rangle\}$.

⋆ $\mathsf{dec}(\Phi_1 \wedge \Phi_2) = \mathsf{dec}(\Phi_1) \otimes \mathsf{dec}(\Phi_2)$, where
$\mathsf{dec}(\Phi_1) \otimes \mathsf{dec}(\Phi_2) := \{\langle\psi_i \wedge \psi_j, \Psi_i \wedge \Psi_j\rangle \mid \langle\psi_i, \Psi_i\rangle \in \mathsf{dec}(\Phi_1), \langle\psi_j, \Psi_j\rangle \in \mathsf{dec}(\Phi_2)\}$.

⋆ $\mathsf{dec}(\Phi_1 \vee \Phi_2) = \mathsf{dec}(\Phi_1) \cup \mathsf{dec}(\Phi_2) \cup (\mathsf{dec}(\Phi_1) \oplus \mathsf{dec}(\Phi_2))$,
where $\mathsf{dec}(\Phi_1) \oplus \mathsf{dec}(\Phi_2) :=$
$\{\langle\psi_i \wedge \psi_j, \Psi_i \vee \Psi_j\rangle \mid \langle\psi_i, \Psi_i\rangle \in \mathsf{dec}(\Phi_1),\ \langle\psi_j, \Psi_j\rangle \in \mathsf{dec}(\Phi_2),\ \Psi_i \neq \top, \Psi_j \neq \top\}$.

The conjunctive case is clear: every path satisfying $\Phi_1 \wedge \Phi_2$ combines a type of path satisfying Φ_1 with a type of path satisfying Φ_2. To understand the disjunctive case, first note that the use of $\mathsf{dec}(\Phi_1) \oplus \mathsf{dec}(\Phi_2)$ in the above union reflects the case of those plays where it is not decided yet which disjunct of $\Phi_1 \vee \Phi_2$ will hold, so we have to keep both disjuncts true at the present state and delay the choice. This is why the state formulae ψ_i and ψ_j are connected by $\wedge$ but the path formulae Ψ_i and Ψ_j are connected by $\vee$. Moreover, the $\oplus$ operation avoids the construction of a pair $\langle\psi_i \wedge \psi_j, \Psi_i \vee \Psi_j\rangle$ where either Ψ_i or Ψ_j is $\top$, because in that case we would be in a situation already included in $\mathsf{dec}(\Phi_1)$ or in $\mathsf{dec}(\Phi_2)$. The three cases for paths satisfying the disjunction $\Phi_1 \vee \Phi_2$ can be illustrated by the picture in Figure 1.

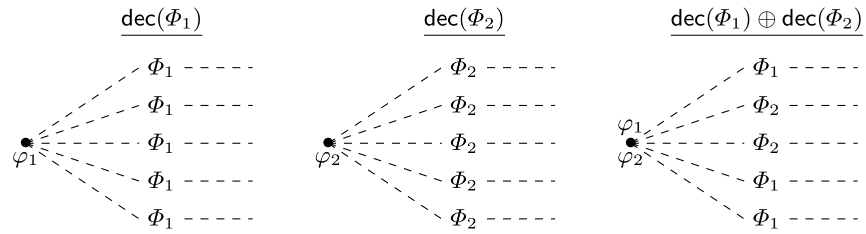

Fig. 1. The 3 cases for disjunctive path objectives in a γ-formula.

Now, let $\zeta = \langle\langle A\rangle\rangle\Phi$ or $\zeta = [\![A]\!]\Phi$ be a γ-formula to be decomposed. Each pair $\langle\psi, \Psi\rangle \in \mathsf{dec}(\Phi)$ is then converted to a *γ-component* $\gamma(\psi, \Psi)$ as follows:

$$\gamma(\psi, \Psi) = \psi \quad \text{if } \Psi = \top \tag{5}$$

$$\gamma(\psi, \Psi) = \psi \wedge \langle\langle A\rangle\rangle \bigcirc \langle\langle A\rangle\rangle\Psi \quad \text{if } \zeta \text{ is of the form } \langle\langle A\rangle\rangle\Phi, \tag{6}$$

$$\gamma(\psi, \Psi) = \psi \wedge [\![A]\!] \bigcirc [\![A]\!]\Psi \quad \text{if } \zeta \text{ is of the form } [\![A]\!]\Phi \tag{7}$$

The following key lemma claims that every γ-formula is equivalent to the disjunction of its γ-components. For the (long and non-trivial) proof see [2].

Lemma 1. *For any* ATL^+ *γ-formula* $\Theta = \langle\langle A\rangle\rangle\Phi$ *or* $\Theta = [\![A]\!]\Phi$*:*

1. $\Phi \equiv \bigvee\{\psi \wedge \bigcirc\Psi \mid \langle\psi, \Psi\rangle \in \mathsf{dec}(\Phi)\}$.
2. $\langle\langle A\rangle\rangle\Phi \equiv \bigvee\{\langle\langle A\rangle\rangle(\psi \wedge \bigcirc\Psi) \mid \langle\psi, \Psi\rangle \in \mathsf{dec}(\Phi)\}$, *and respectively,* $[\![A]\!]\Phi \equiv \bigvee\{[\![A]\!](\psi \wedge \bigcirc\Psi) \mid \langle\psi, \Psi\rangle \in \mathsf{dec}(\Phi)\}$.
3. $\Theta \equiv \bigvee\{\gamma(\psi, \Psi) \mid \langle\psi, \Psi\rangle \in \mathsf{dec}(\Phi)\}$.

Example 1. We will use 2 syntactically similar, yet different, running examples: $\theta = \langle\langle 1\rangle\rangle(p\mathcal{U}q \vee \Box q) \wedge \langle\langle 2\rangle\rangle(\Diamond p \wedge \Box\neg q)$ and $\vartheta = \langle\langle 1\rangle\rangle(p\mathcal{U}q \vee \Box q) \wedge [\![2]\!](\Diamond p \wedge \Box\neg q)$.

First, we consider θ. It is an α-formula with conjunctive components $\theta_1 = \langle\langle 1\rangle\rangle(p\mathcal{U}q \vee \Box q)$ and $\theta_2 = \langle\langle 2\rangle\rangle(\Diamond p \wedge \Box\neg q)$. Further, θ_1 is a γ-formula of the form $\langle\langle A\rangle\rangle\Phi$ where the main connective of Φ is $\vee$. So $\mathsf{dec}(\theta_1) = \mathsf{dec}(p\mathcal{U}q) \cup \mathsf{dec}(\Box q) \cup (\mathsf{dec}(p\mathcal{U}q) \oplus \mathsf{dec}(\Box q))$, where $\mathsf{dec}(p\mathcal{U}q) = \{\langle p, p\mathcal{U}q\rangle, \langle q, \top\rangle\}$ and $\mathsf{dec}(\Box q) = \{\langle q, \Box q\rangle\}$. Thus, $\mathsf{dec}(\theta_1) = \{\langle p, p\mathcal{U}q\rangle, \langle q, \top\rangle, \langle q, \Box q\rangle, \langle p \wedge q, p\mathcal{U}q \vee \Box q\rangle\}$, hence $\theta_1 \equiv (p \wedge \langle\langle 1\rangle\rangle\bigcirc\langle\langle 1\rangle\rangle p\mathcal{U}q) \vee (q) \vee (q \wedge \langle\langle 1\rangle\rangle\bigcirc\langle\langle 1\rangle\rangle\Box q) \vee (p \wedge q \wedge \langle\langle 1\rangle\rangle\bigcirc\langle\langle 1\rangle\rangle(p\mathcal{U}q \vee \Box q))$. Likewise, θ_2 is a γ-formula of the form $\langle\langle A\rangle\rangle\Phi$ where the main connective of Φ is $\wedge$. So $\mathsf{dec}(\theta_2) = \mathsf{dec}(\Diamond p) \otimes \mathsf{dec}(\Box\neg q)$, with $\mathsf{dec}(\Diamond p) = \{\langle T, \Diamond p\rangle, \langle p, T\rangle\}$ and $\mathsf{dec}(\Box\neg q) = \{\langle\neg q, \Box\neg q\rangle\}$. Thus, $\mathsf{dec}(\theta_2) = \{\langle\top \wedge \neg q, \Diamond p \wedge \Box\neg q\rangle, \langle p \wedge \neg q, \top \wedge \Box\neg q\rangle\} = \{\langle\neg q, \Diamond p \wedge \Box\neg q\rangle, \langle p \wedge \neg q, \Box\neg q\rangle\}$ and $\theta_2 \equiv (\neg q \wedge \langle\langle 2\rangle\rangle\bigcirc\langle\langle 2\rangle\rangle(\Diamond p \wedge \Box\neg q)) \vee (p \wedge \neg q \wedge \langle\langle 2\rangle\rangle\bigcirc\langle\langle 2\rangle\rangle\Box\neg q)$.

For ϑ, the γ-decomposition is similar, we only replace $\langle\langle 2\rangle\rangle$ by $[\![2]\!]$. Thus, we obtain $\vartheta_1 \equiv (p \wedge \langle\langle 1\rangle\rangle\bigcirc\langle\langle 1\rangle\rangle p\mathcal{U}q) \vee (q) \vee (q \wedge \langle\langle 1\rangle\rangle\bigcirc\langle\langle 1\rangle\rangle\Box q) \vee (p \wedge q \wedge \langle\langle 1\rangle\rangle\bigcirc\langle\langle 1\rangle\rangle(p\mathcal{U}q \vee \Box q))$ and $\vartheta_2 \equiv (\neg q \wedge [\![2]\!]\bigcirc[\![2]\!](\Diamond p \wedge \Box\neg q)) \vee (p \wedge \neg q \wedge [\![2]\!]\bigcirc[\![2]\!]\Box\neg q)$.

The *closure* $cl(\psi)$ of an ATL^+ state formula ψ is the least set of ATL^+ formulae such that $\psi, \top, \bot \in cl(\psi)$ and $cl(\psi)$ is closed under taking of successor-, α-, β- and γ-components. For any set of state formulae Γ we define $cl(\Gamma) := \bigcup\{cl(\psi) \mid \psi \in \Gamma\}$. We denote by $|\psi|$ the length of ψ and by $\|\Gamma\|$ the cardinality of Γ.

Example 2. The closures of the formulae θ and ϑ from Example 1 are:

$\mathsf{cl}(\theta) = \{\theta, \theta_1, \theta_2, p \wedge q \wedge \langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle(p\mathcal{U}q \vee \Box q), p \wedge q, p, q, \langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle(p\mathcal{U}q \vee \Box q), q \wedge \langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle \Box q, \langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle \Box q, \langle\langle 1\rangle\rangle \Box q, p \wedge \langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle p\mathcal{U}q, \langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle p\mathcal{U}q, \langle\langle 1\rangle\rangle p\mathcal{U}q, p \wedge \neg q \wedge \langle\langle 2\rangle\rangle \bigcirc \langle\langle 2\rangle\rangle \Box\neg q, p \wedge \neg q, \neg q, \langle\langle 2\rangle\rangle \bigcirc \langle\langle 2\rangle\rangle \Box\neg q, \langle\langle 2\rangle\rangle \Box\neg q, \neg q \wedge \langle\langle 2\rangle\rangle \bigcirc \langle\langle 2\rangle\rangle \Box\neg q, \neg q \wedge \langle\langle 2\rangle\rangle \bigcirc \langle\langle 2\rangle\rangle(\Diamond p \wedge \Box\neg q), \langle\langle 2\rangle\rangle \bigcirc \langle\langle 2\rangle\rangle(\Diamond p \wedge \Box\neg q), \top\}$.

$\mathsf{cl}(\vartheta) = \{\vartheta, \vartheta_1, \vartheta_2, p \wedge q \wedge \langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle(p\mathcal{U}q \vee \Box q), p \wedge q, p, q, \langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle(p\mathcal{U}q \vee \Box q), q \wedge \langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle \Box q, \langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle \Box q, \langle\langle 1\rangle\rangle \Box q, p \wedge \langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle p\mathcal{U}q, \langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle p\mathcal{U}q, \langle\langle 1\rangle\rangle p\mathcal{U}q, p \wedge \neg q \wedge [\![2]\!] \bigcirc [\![2]\!] \Box\neg q, p \wedge \neg q, \neg q, [\![2]\!] \bigcirc [\![2]\!] \Box\neg q, [\![2]\!] \Box\neg q, \neg q \wedge [\![2]\!] \bigcirc [\![2]\!] \Box\neg q, \neg q \wedge [\![2]\!] \bigcirc [\![2]\!](\Diamond p \wedge \Box\neg q), [\![2]\!] \bigcirc [\![2]\!](\Diamond p \wedge \Box\neg q), \top\}$.

Lemma 2. *For any* ATL$^+$ *state formula* φ, $\|cl(\varphi)\| < 2^{|\varphi|^2}$.

Proof. Every formula in $cl(\varphi)$ has length less than $2|\varphi|$ and is built from symbols in φ, so there can be at most $|\varphi|^{2|\varphi|} = 2^{2|\varphi| \log_2 |\varphi|} < 2^{|\varphi|^2}$ such formulae. □

The estimate above is rather crude, but $\|cl(\varphi)\|$ *can* reach size exponential in $|\varphi|$. Indeed, consider the formulae $\phi_k = \langle\langle 1\rangle\rangle(p_1 \mathcal{U} q_1 \wedge (p_2 \mathcal{U} q_2 \wedge (\ldots \wedge p_k \mathcal{U} q_k) \ldots)$ for $k = 1, 2, \ldots$ and distinct $p_1, q_1, \ldots, p_k, q_k, \ldots \in \mathsf{Prop}$. Then $|\phi_k| = O(k)$, while the number of different γ-components of ϕ_k is 2^k, hence $\|cl(\phi_k)\| > 2^k$.

3.2 Full expansions of sets of ATL$^+$ formulae

As part of the tableaux construction we will need a procedure that, for any given finite set of ATL$^+$ state formulae Γ, produces all "full expansions" (called in [7] "downward saturated extensions") defined below.

Definition 1. *Let* Γ, Δ *be sets of* ATL$^+$ *state formulae and* $\Gamma \subseteq \Delta \subseteq cl(\Gamma)$.

1. Δ *is* patently inconsistent *if it contains* $\bot$ *or a pair of formulae* φ *and* $\neg\varphi$.
2. Δ *is a* full expansion of Γ *if it is not patently inconsistent and satisfies the following closure conditions:*
   - *if* $\varphi \wedge \psi \in \Delta$ *then* $\varphi \in \Delta$ *and* $\psi \in \Delta$;
   - *if* $\varphi \vee \psi \in \Delta$ *then* $\varphi \in \Delta$ *or* $\psi \in \Delta$;
   - *if* $\varphi \in \Delta$ *is a* γ*-formula, then at least one* γ*-component of* φ *is in* Δ *and exactly one of these* γ*-components in* Δ*, denoted* $\gamma(\varphi, \Delta)$*, is designated as* the γ-component in Δ linked to the γ-formula φ*, as explained below.*

The family of all full expansions of Γ will be denoted by $FE(\Gamma)$. It can be constructed by a simple iterative procedure that starts with $\{\Gamma\}$ and repeatedly, until saturation, takes a set X from the currently constructed family, selects a formula $\varphi \in X$ and: if φ is a conjunction, then adds both conjunctive components of φ to X; if φ is a disjunction, then creates two extensions of X by adding respectively each disjunctive component of φ; and if φ is a γ-formula, then creates an extension of X with each γ-component ψ of φ and designates ψ as the γ-component of φ linked to φ in every full expansion of Γ eventually produced by further extending $X \cup \{\psi\}$. In case when such an extension becomes patently inconsistent it is discarded from the family. Clearly, this procedure terminates on every finite input set of formulae Γ and produces a family of at most $2^{\|cl(\Gamma)\|}$ sets. Furthermore, due to Lemma 1, we have the following:

Proposition 1. *For any finite set of* ATL$^+$ *state formulae* Γ*:*

$$\bigwedge \Gamma \equiv \bigvee \left\{ \bigwedge \Delta \mid \Delta \in FE(\Gamma) \right\}.$$

4 Tableau-based decision procedure for ATL$^+$

The tableaux procedure consists of three major phases: *pretableau construction*, *prestate elimination*, and *state elimination*. It constructs a directed graph $\mathcal{T}^\eta$ (called a *tableau*) with nodes labelled by finite sets of formulae and directed edges between nodes relating them to successor nodes.

The pretableau construction phase produces the so-called *pretableau* $\mathcal{P}^\eta$ for the input formula η, with two kinds of nodes: *states* and *prestates*. States are fully expanded sets, meant to represent states of a CGM, while prestates can be any finite sets of formulae from $cl(\eta)$ and only play a temporary role in the construction of $\mathcal{P}^\eta$. States and prestates are labelled uniquely, so they can be identified with their labels. The prestate elimination phase creates a smaller graph $\mathcal{T}_0^\eta$ out of $\mathcal{P}^\eta$, called the *initial tableau for* η, by eliminating all the prestates from $\mathcal{P}^\eta$ and accordingly redirecting its edges. Finally, the state elimination phase removes, step-by-steps, all the states (if any) that cannot be satisfied in a CGM, because they lack necessary successors or because they contain unrealized eventualities. Eventually, the elimination procedure produces a (possibly empty) subgraph $\mathcal{T}^\eta$ of $\mathcal{T}_0^\eta$, called the *final tableau for* η. If some state Δ of $\mathcal{T}^\eta$ contains η, the procedure declares η satisfiable and a partly defined CGM (called *Hintikka game frame*) satisfying η can be extracted from it; otherwise it declares η unsatisfiable.

4.1 Pretableau construction phase

The pretableau construction phase for an input formula η starts with an initial prestate (with label) $\{\eta\}$ and consists of alternating application of two construction rules, until saturation: **(SR)**, expanding prestates into states, and **(Next)**, creating successor prestates from states. This phase closely resembles the corresponding one for ATL tableaux in [7], with the only essential difference being the γ-decomposition of γ-formulae used here by the rule **(SR)**, which causes, as we will see, a possibly exponential blow-up of the size of the tableaux, and eventually of the entire worst case time complexity, as compared to the ATL tableaux. Another (minor) difference with respect to [7] is in the formulation of both rules, because here we work with formulae in negation normal form.

Rule (SR) Given a prestate Γ, do the following:

1. For each full expansion Δ of Γ add to the pretableau a state with label Δ.
2. For each of the added states Δ, if Δ does not contain any formulae of the form $\langle\langle A\rangle\rangle \bigcirc \varphi$ or $[\![A]\!] \bigcirc \varphi$, add the formula $\langle\langle \mathbb{A}\rangle\rangle \bigcirc \top$ to it;
3. For each state Δ obtained at steps 1 and 2, link Γ to Δ via a $\Longrightarrow$ edge;
4. If, however, the pretableau already contains a state Δ' with label Δ, do not create another copy of it but only link Γ to Δ' via a $\Longrightarrow$ edge.

Example 3. For the formula θ from Example 1 the initial prestate is $\Gamma_0 = \{\langle\langle 1\rangle\rangle(p\mathcal{U}q \vee \Box q) \wedge \langle\langle 2\rangle\rangle(\Diamond p \wedge \Box\neg q)\}$. It has 2 full expansions:
$\Delta_1 = \{\theta, \theta_1, \theta_2, p, \neg q, \langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle p\mathcal{U}q, \langle\langle 2\rangle\rangle \bigcirc \langle\langle 2\rangle\rangle(\Diamond p \wedge \Box\neg q)\}$, and
$\Delta_2 = \{\theta, \theta_1, \theta_2, p, p \wedge \neg q, \neg q, \langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle p\mathcal{U}q, \langle\langle 2\rangle\rangle \bigcirc \langle\langle 2\rangle\rangle \Box\neg q\}$.

Likewise, for the formula ϑ: $\Gamma_0 = \{\langle\langle 1\rangle\rangle(p\mathcal{U}q \vee \Box q) \wedge [\![2]\!](\Diamond p \wedge \Box\neg q)\}$ is the initial prestate and it has 2 full expansions:
$\Delta_1 = \{\vartheta, \vartheta_1, \vartheta_2, p, \neg q, \langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle p\mathcal{U}q, [\![2]\!] \bigcirc [\![2]\!](\Diamond p \wedge \Box\neg q)\}$, and
$\Delta_2 = \{\vartheta, \vartheta_1, \vartheta_2, p, p \wedge \neg q, \neg q, \langle\langle 1\rangle\rangle \bigcirc \langle\langle 1\rangle\rangle p\mathcal{U}q, [\![2]\!] \bigcirc [\![2]\!] \Box\neg q\}$.

In the following, by *enforceable successor formula* we mean a formula of the form $\langle\langle A\rangle\rangle \bigcirc \psi$ and by *unavoidable successor formula* one of the form $[\![A]\!] \bigcirc \psi$.

Rule (Next) Given a state Δ do the following, where σ is a shorthand for $\sigma_{\mathbb{A}}$:

1. List all primitive successor formulae of Δ in such a way that all enforceable successor formulae precede all unavoidable ones; let the result be the list

$$\mathbb{L} = \langle\langle A_0\rangle\rangle \bigcirc \varphi_0, \ldots, \langle\langle A_{m-1}\rangle\rangle \bigcirc \varphi_{m-1}, [\![A'_0]\!] \bigcirc \psi_0, \ldots, [\![A'_{l-1}]\!] \bigcirc \psi_{l-1}$$

   Let $r_\Delta = m + l$; denote by $D(\Delta)$ the set $\{0, \ldots, r_\Delta - 1\}^{|\mathbb{A}|}$. Then, for every $\sigma \in D(\Delta)$, denote $N(\sigma) := \{i \mid \sigma_i \geqslant m\}$, where σ_i is the ith component of the tuple σ, and let $\mathbf{co}(\sigma) := [\Sigma_{i \in N(\sigma)}(\sigma_i - m)] \mod l$.
2. For each $\sigma \in D(\Delta)$ create a prestate:

$$\begin{aligned}\Gamma_\sigma = \{\varphi_p \mid \langle\langle A_p\rangle\rangle \bigcirc \varphi_p \in \Delta \text{ and } \sigma_a = p \text{ for all } a \in A_p\}\\ \cup \{\psi_q \mid [\![A'_q]\!] \bigcirc \psi_q \in \Delta, \mathbf{co}(\sigma) = q, \text{ and } \mathbb{A} - A'_q \subseteq N(\sigma)\}\end{aligned}$$

   If Γ_σ is empty, add $\top$ to it. Then connect Δ to Γ_σ with $\xrightarrow{\sigma}$.
   If, however, $\Gamma_\sigma = \Gamma$ for some prestate Γ that has already been added to the pretableau, only connect Δ to Γ with $\xrightarrow{\sigma}$.

For intuition on the rule **(Next)** see [7] and [2]. The rules **(SR)** and **(Next)** are applied alternatively until saturation, which is bound to occur because every label is a subset of $cl(\eta)$. Then the construction phase is over. The graph built in that phase is called *pretableau* for the input formula η and denoted by $\mathcal{P}^\eta$.

Example 4. Continuation of Example 3 for θ: For Δ_1, the list of successor formulae is $\mathbb{L} = \langle\langle 1\rangle\rangle\bigcirc\langle\langle 1\rangle\rangle p\mathcal{U}q, \langle\langle 2\rangle\rangle\bigcirc\langle\langle 2\rangle\rangle(\Diamond p \wedge \Box\neg q)$, so $m = 2$, $l = 0$ and $r_{\Delta_1} = 2$. As there are no unavoidable successor formulae, we do not need to compute $N(\sigma)$ and $\mathbf{co}(\sigma)$. Then, $\Gamma_{(0,0)} = \{\langle\langle 1\rangle\rangle p\mathcal{U}q\} = \Gamma_1$, $\Gamma_{(0,1)} = \{\langle\langle 1\rangle\rangle p\mathcal{U}q, \langle\langle 2\rangle\rangle(\Diamond p \wedge \Box\neg q)\} = \Gamma_2$, $\Gamma_{(1,0)} = \{\top\} = \Gamma_3$ and $\Gamma_{(1,1)} = \{\langle\langle 2\rangle\rangle(\Diamond p \wedge \Box\neg q)\} = \Gamma_4$.

For Δ_2, the list of successor formulae is $\mathbb{L} = \langle\langle 1\rangle\rangle\bigcirc\langle\langle 1\rangle\rangle p\mathcal{U}q, \langle\langle 2\rangle\rangle\bigcirc\langle\langle 2\rangle\rangle\Box\neg q$, so $m = 2$, $l = 0$ and $r_{\Delta_2} = 2$. Here again, we do not compute $N(\sigma)$ and $\mathbf{co}(\sigma)$. Then $\Gamma_{(0,0)} = \{\langle\langle 1\rangle\rangle p\mathcal{U}q\} = \Gamma_1$, $\Gamma_{(0,1)} = \{\langle\langle 1\rangle\rangle p\mathcal{U}q, \langle\langle 2\rangle\rangle\Box\neg q\} = \Gamma_5$, $\Gamma_{(1,0)} = \{\top\} = \Gamma_3$ and $\Gamma_{(1,1)} = \{\langle\langle 2\rangle\rangle\Box\neg q\} = \Gamma_6$.

Applying rule **(SR)** to the so-obtained prestates, we have:
$\mathbf{states}(\Gamma_1) = \{\Delta_3 : \{\langle\langle 1\rangle\rangle p\mathcal{U}q, p, \langle\langle 1\rangle\rangle\bigcirc\langle\langle 1\rangle\rangle p\mathcal{U}q\}, \Delta_4 : \{\langle\langle 1\rangle\rangle p\mathcal{U}q, q, \langle\langle 1, 2\rangle\rangle \bigcirc \top\}\}$,
$\mathbf{states}(\Gamma_2) = \Delta_5 : \{\langle\langle 1\rangle\rangle p\mathcal{U}q, \langle\langle 2\rangle\rangle(\Diamond p \wedge \Box\neg q), p, \neg q, \langle\langle 1\rangle\rangle\bigcirc\langle\langle 1\rangle\rangle p\mathcal{U}q, \langle\langle 2\rangle\rangle\bigcirc\langle\langle 2\rangle\rangle(\Diamond p \wedge$

$\Box\neg q)\}$, $\Delta_6 : \{\langle\langle 1\rangle\rangle p\mathcal{U}q, \langle\langle 2\rangle\rangle(\Diamond p \wedge \Box\neg q), p, p\wedge\neg q, \neg q, \langle\langle 1\rangle\rangle\bigcirc\langle\langle 1\rangle\rangle p\mathcal{U}q, \langle\langle 2\rangle\rangle\bigcirc\langle\langle 2\rangle\rangle\Box\neg q\}\}$;
states$(\Gamma_3) = \{\Delta_7 : \{\top, \langle\langle 1,2\rangle\rangle \bigcirc \top\}\}$;
states$(\Gamma_4) = \{\Delta_8 : \{\langle\langle 2\rangle\rangle(\Diamond p \wedge \Box\neg q), \neg q, \langle\langle 2\rangle\rangle\bigcirc\langle\langle 2\rangle\rangle(\Diamond p \wedge \Box\neg q)\}, \Delta_9 : \{\langle\langle 2\rangle\rangle(\Diamond p \wedge \Box\neg q), p \wedge \neg q, \neg q, \langle\langle 2\rangle\rangle\bigcirc\langle\langle 2\rangle\rangle\Box\neg q\}\}$;
states$(\Gamma_5) = \{\Delta_{10} : \{\langle\langle 1\rangle\rangle p\mathcal{U}q, \langle\langle 2\rangle\rangle\Box\neg q, p, \neg q, \langle\langle 1\rangle\rangle\bigcirc\langle\langle 1\rangle\rangle p\mathcal{U}q, \langle\langle 2\rangle\rangle\bigcirc\langle\langle 2\rangle\rangle\Box\neg q\}\}$;
states$(\Gamma_6) = \{\Delta_{11} : \{\langle\langle 2\rangle\rangle\Box\neg q, \neg q, \langle\langle 2\rangle\rangle\bigcirc\langle\langle 2\rangle\rangle\Box\neg q\}\}$.

The pretableau for θ is given in Figure 2.

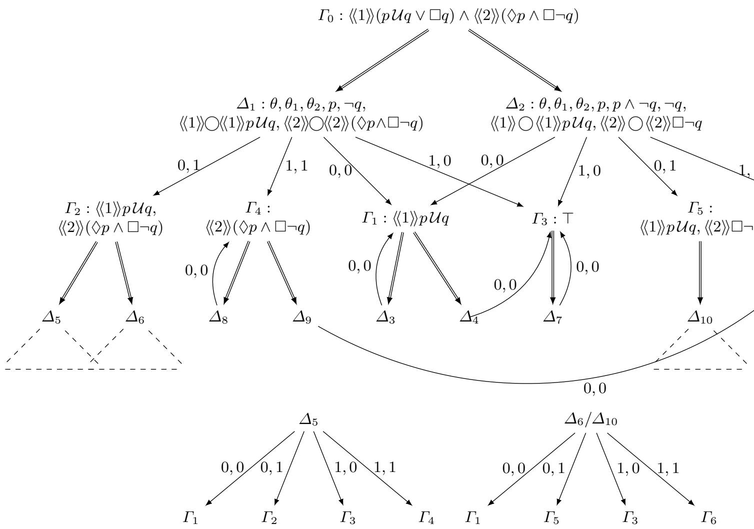

Fig. 2. The pretableau for θ

Example 5. Continuation of Example 3 for ϑ: For Δ_1, the list of successor formulae is $\mathbb{L} = \langle\langle 1\rangle\rangle\bigcirc\langle\langle 1\rangle\rangle p\mathcal{U}q, [\![2]\!]\bigcirc[\![2]\!](\Diamond p \wedge \Box\neg q)$, so $m = 1$, $l = 1$ and $r_{\Delta_1} = 2$. Therefore, $N(0,0) = \emptyset$, $N(0,1) = \{2\}$, $N(1,0) = \{1\}$, $N(1,1) = \{1,2\}$ and also $\mathbf{co}(0,0) = \mathbf{co}(0,1) = 0 = \mathbf{co}(0,1) = \mathbf{co}(0,1) = 0$. Then, $\Gamma_{(0,0)} = \Gamma_{(0,1)} = \{\langle\langle 1\rangle\rangle p\mathcal{U}q\} = \Gamma_1$, and $\Gamma_{(1,0)} = \Gamma_{(1,1)} = \{[\![2]\!](\Diamond p \wedge \Box\neg q)\} = \Gamma_2$.

For Δ_2, the list of successor formulae is $\mathbb{L} = \langle\langle 1\rangle\rangle\bigcirc\langle\langle 1\rangle\rangle p\mathcal{U}q, [\![2]\!]\bigcirc[\![2]\!]\Box\neg q$, so $m = 1$, $l = 1$ and $r_{\Delta_2} = 2$. Here also $N(0,0) = \emptyset$, $N(0,1) = \{2\}$, $N(1,0) = \{1\}$, $N(1,1) = \{1,2\}$, and $\mathbf{co}(0,0) = \mathbf{co}(0,1) = \mathbf{co}(0,1) = \mathbf{co}(0,1) = 0$. Then, $\Gamma_{(0,0)} = \Gamma_{(0,1)} = \{\langle\langle 1\rangle\rangle p\mathcal{U}q\} = \Gamma_1$, and $\Gamma_{(1,0)} = \Gamma_{(1,1)} = \{[\![2]\!]\Box\neg q\} = \Gamma_3$.

In the same way, we obtain:
states$(\Gamma_1) = \{\Delta_3 : \{\langle\langle 1\rangle\rangle p\mathcal{U}q, p, \langle\langle 1\rangle\rangle\bigcirc\langle\langle 1\rangle\rangle p\mathcal{U}q\}, \Delta_4 : \{\langle\langle 1\rangle\rangle p\mathcal{U}q, q, \langle\langle 1,2\rangle\rangle\bigcirc\top\}\}$;
states$(\Gamma_2) = \{\Delta_5 : \{[\![2]\!](\Diamond p \wedge \Box\neg q), \neg q, [\![2]\!]\bigcirc[\![2]\!](\Diamond p \wedge \Box\neg q)\}, \Delta_6 : \{[\![2]\!](\Diamond p \wedge \Box\neg q), p\wedge\neg q, p, \neg q, [\![2]\!]\bigcirc[\![2]\!]\Box\neg q\}\}$; **states**$(\Gamma_3) = \{\Delta_7 : \{[\![2]\!]\Box\neg q, \neg q, [\![2]\!]\bigcirc[\![2]\!]\Box\neg q\}\}$.

4.2 The prestate and state elimination phases. Eventualities

First, we remove from $\mathcal{P}^\eta$ all the prestates and the $\Longrightarrow$ edges, as follows. For every prestate Γ in $\mathcal{P}^\eta$ put $\Delta \xrightarrow{\sigma} \Delta'$ for all states Δ in $\mathcal{P}^\eta$ with $\Delta \xrightarrow{\sigma} \Gamma$ and all

$\Delta' \in \mathbf{states}(\Gamma)$; then, remove Γ from $\mathcal{P}^\eta$. The graph obtained after eliminating all prestates is called the *initial tableau*, denoted by $\mathcal{T}_0^\eta$. The initial tableau for the formula θ in our running example is given on Figure 3.

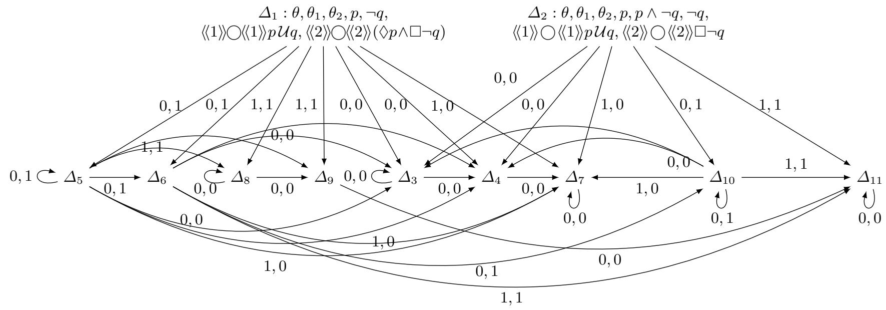

Fig. 3. The initial tableau for θ

The elimination phase starts with $\mathcal{T}_0^\eta$ and goes through stages. At stage $n+1$ we remove exactly one state from the tableau $\mathcal{T}_n^\eta$ obtained at the previous stage, by applying one of the elimination rules described below, thus obtaining the tableau $\mathcal{T}_{n+1}^\eta$. The set of states of $\mathcal{T}_m^\eta$ is noted S_m^η.

The first elimination rule **(ER1)**, defined below, is used to eliminate all states with missing successors for some move vectors determined by the rule **(Next)**. If, due to a previous state elimination, any state has an outgoing move vector for which the corresponding successor state is missing, we delete the state. The reason is clear: if Δ is to be satisfiable, then for each $\sigma \in D(\Delta)$ there should exist a satisfiable Δ' that Δ reaches via σ. Formally, the rule is stated as follows, where $D(\Delta)$ is defined in the rule **(Next)**:

Rule (ER1): If, for some $\sigma \in D(\Delta)$, all states Δ' with $\Delta \xrightarrow{\sigma} \Delta'$ have been eliminated at earlier stages, then obtain $\mathcal{T}_{n+1}^\eta$ by eliminating Δ from $\mathcal{T}_n^\eta$.

The aim of the next elimination rule is to make sure that there are no *unrealized eventualities*. In ATL there are only two kinds of eventualities : $\langle\langle A\rangle\rangle\varphi\mathcal{U}\psi$ and $[\![A]\!]\Diamond\varphi$. The situation is more complex in ATL$^+$. For instance, should the formula $\langle\langle A\rangle\rangle(\Box\varphi \vee \psi_1\mathcal{U}\psi_2)$ be considered an eventuality? Our solution for ATL$^+$ is to consider all γ-formulae as *potential eventualities*. In order to properly define the notion of *realization of a potential eventuality* we first define a function *Real* associating to a γ-formula ψ and a set of ATL$^+$-formulae (representing a state of the current tableau) a Boolean value indicating whether the potential eventuality represented by ψ has been 'realized' at that state:

- $Real(\Phi \wedge \Psi, \Theta) = Real(\Phi, \Theta) \wedge Real(\Psi, \Theta)$
- $Real(\Phi \vee \Psi, \Theta) = Real(\Phi, \Theta) \vee Real(\Psi, \Theta)$
- $Real(\varphi, \Theta) = true$ if $\varphi \in \Theta$, $false$ otherwise
- $Real(\bigcirc\varphi, \Theta) = false$

– $Real(\Box\varphi, \Theta) = true$ if $\varphi \in \Theta$, $false$ otherwise
– $Real(\varphi\mathcal{U}\psi, \Theta) = true$ if $\psi \in \Theta$, $false$ otherwise

Definition 2 (Descendant potential eventualities). *Let $\xi \in \Delta$ be a potential eventuality of the form $\langle\langle A\rangle\rangle\Phi$ or $[\![A]\!]\Phi$. Suppose the γ-component $\gamma(\xi, \Delta)$ in Δ linked to ξ is, respectively, of the form $\psi \wedge \langle\langle A\rangle\rangle \bigcirc \langle\langle A\rangle\rangle\Psi$ or $\psi \wedge [\![A]\!] \bigcirc [\![A]\!]\Psi$. Then the* successor potential eventuality *of ξ w.r.t. $\gamma(\xi, \Delta)$ is the γ-formula $\langle\langle A\rangle\rangle\Psi$ (resp. $[\![A]\!]\Psi$) and it will be denoted by ξ^1_Δ. The notion of* descendant potential eventuality of ξ of degree d, *for $d > 1$, is defined inductively as follows:*

- any successor eventuality of ξ (w.r.t. some γ-component of ξ) is a descendant eventuality of ξ of degree 1;

- any successor eventuality of a descendant eventuality ξ^n of ξ of degree n is a descendant eventuality of ξ of degree $n + 1$.

We will also consider ξ to be a descendant eventuality of itself of degree 0.

Example 6. (*Continuation of Example 5*) In Δ_1 we have $\xi = \langle\langle 1\rangle\rangle(p\mathcal{U}q \vee \Box q)$ with $Real(p\mathcal{U}q \vee \Box q, \Delta_1) = Real(p\mathcal{U}q, \Delta_1) \vee Real(\Box q, \Delta_1) = false \vee false = false$, since $q \notin \Delta_1$, and $\xi' = \langle\langle 2\rangle\rangle(\Diamond p \wedge \Box\neg q)$ with $Real(\Diamond p \wedge \Box\neg q, \Delta_1) = Real(\Diamond p, \Delta_1) \wedge Real(\Box\neg q, \Delta_1) = true \wedge true = true$ since $p, \neg q \in \Delta_1$.

The successor eventuality of $\xi = \langle\langle 1\rangle\rangle(p\mathcal{U}q \vee \Box q)$ w.r.t $\gamma(\xi, \Delta_5)$ is $\xi^1_{\Delta_5} = \langle\langle 1\rangle\rangle p\mathcal{U}q$ in $\Delta_3, \Delta_4, \Delta_5, \Delta_6$. For each $n > 1$, the descendant eventuality of degree n of ξ w.r.t $\gamma(\xi, \Delta_5)$ is $\xi^n_{\Delta_5} = \xi^1_{\Delta_5}$ in $\Delta_3, \Delta_4, \Delta_5, \Delta_6, \Delta_{10}$. The successor eventuality of $\xi' = \langle\langle 2\rangle\rangle(\Diamond p \wedge \Box\neg q)$ w.r.t $\gamma(\xi', \Delta_5)$ is $\xi'^1_{\Delta_5} = \langle\langle 2\rangle\rangle(\Diamond p \wedge \Box\neg q)$ in Δ_5, Δ_6, Δ_8. For each $n > 1$, the descendant eventualities of degree n of ξ' w.r.t $\gamma(\xi', \Delta_5)$ are $\xi'^n_{\Delta_5} = \xi'^1_{\Delta_5}$ in Δ_5, Δ_6, Δ_8 and Δ_9; and $\xi'^n_{\Delta_5} = \langle\langle 2\rangle\rangle\Box\neg q$ in Δ_{10} and Δ_{11}.

Now, let $\mathbb{L} = \langle\langle A_0\rangle\rangle \bigcirc \varphi_0, \ldots, \langle\langle A_{m-1}\rangle\rangle \bigcirc \varphi_{m-1}, [\![A'_0]\!] \bigcirc \psi_0, \ldots, [\![A'_{l-1}]\!] \bigcirc \psi_{l-1}$ be the list of all primitive successor formulae of $\Delta \in S^\eta_0$, induced as part of application of **(Next)**. We will use the following notation:
$D(\Delta, \langle\langle A_p\rangle\rangle \bigcirc \varphi) := \{\sigma \in D(\Delta) \mid \sigma_a = p \text{ for every } a \in A_p\}$
$D(\Delta, [\![A'_q]\!] \bigcirc \psi) := \{\sigma \in D(\Delta) \mid \mathbf{co}(\sigma) = q \text{ and } \mathbb{A} - A'_q \subseteq N(\sigma)\}$

Next, we will define recursively what it means for an eventuality ξ to be realized at a state Δ of a tableau $\mathcal{T}^\eta_n$, followed by our second elimination rule.

Definition 3 (Realization of potential eventualities). *Let $\Delta \in S^\eta_n$ and $\xi \in \Delta$ be a potential eventuality of the form $\langle\langle A\rangle\rangle\Phi$ or $[\![A]\!]\Phi$. Then:*

1. *If $Real(\Phi, \Delta) = true$ then ξ is realized at Δ in $\mathcal{T}^\eta_n$.*
2. *Else, let ξ^1_Δ be the successor potential eventuality of ξ w.r.t. $\gamma(\xi, \Delta)$. If for every $\sigma \in D(\Delta, \langle\langle A\rangle\rangle \bigcirc \xi^1_\Delta)$ (resp. $\sigma \in D(\Delta, [\![A]\!] \bigcirc \xi^1_\Delta)$), there exists $\Delta' \in \mathcal{T}^\eta_n$ with $\Delta \xrightarrow{\sigma} \Delta'$ and ξ^1_Δ is realized at Δ' in $\mathcal{T}^\eta_n$, then ξ is realized at Δ in $\mathcal{T}^\eta_n$.*

Rule (ER2): If $\Delta \in S^\eta_n$ contains a potential eventuality that is not realized at $\Delta \in \mathcal{T}^\eta_n$, then obtain $\mathcal{T}^\eta_{n+1}$ by removing Δ from S^η_n.

Example 7. (*Cont. of Example 6*) The potential eventuality $\xi'' = \langle\langle 1\rangle\rangle(p\mathcal{U}q)$ is not realized in Δ_5, so by Rule **(ER2)** we remove the state Δ_5 from $\mathcal{T}^\theta_0$ and obtain the tableau $\mathcal{T}^\theta_1$. The same applies to Δ_6 for ξ'', so we also remove Δ_6

from $\mathcal{T}_1^\theta$ and obtain $\mathcal{T}_2^\theta$ with Rule **(ER2)**. In $\mathcal{T}_2^\theta$ there is no more move vector $(0,1)$ for the state Δ_1, so by Rule **(ER1)** we remove Δ_1 from $\mathcal{T}_2^\theta$ and obtain $\mathcal{T}_3^\theta$. In the same way, Δ_{10} is removed by Rule **(ER2)** and Δ_2 by Rule **(ER1)**.

As for the case of ϑ, it is easy to see that no states get eliminated, so the final tableau is the same as the initial one.

The elimination phase is completed when no more applications of elimination rules are possible. Then we obtain the *final tableau* for η, denoted by $\mathcal{T}^\eta$. It is declared *open* if η belongs to some state in it, otherwise *closed*. The procedure for deciding satisfiability of η returns “No” if $\mathcal{T}^\eta$ is closed, “Yes” otherwise.

Example 8. (*Continuation of Example 7*) At the end of the elimination phase, Δ_1 and Δ_2 are no longer in $\mathcal{T}^\theta$. Thus $\mathcal{T}^\theta$ is closed and we deduce that the formula $\theta = \langle\langle 1\rangle\rangle(p\,\mathcal{U}q \vee \Box q) \wedge \langle\langle 2\rangle\rangle(\Diamond p \wedge \Box\neg q)$ is declared unsatisfiable. The final tableaux for θ is given on Figure 4.

Respectively, the final tableau for ϑ is open, hence ϑ is declared satisfiable. Indeed, a CGM can be extracted from the final tableau.

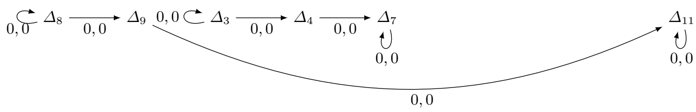

Fig. 4. The final tableau for θ

5 Termination, soundness, completeness and complexity

The termination of the tableaux procedure is straightforward, as there are only finitely many states and prestates that can be added in the construction phase.

Theorem 1. *The tableaux method for* ATL$^+$ *is sound.*

Soundness of the tableaux method means that if the input formula is satisfiable, then the procedure will indeed produce an open tableau. The argument in a nutshell is that if the input formula η is satisfiable, then, due to Proposition 1, there is a satisfiable state Δ in **states**$(\{\eta\})$. The key claim, proved by induction on the number of steps in the elimination phase, is that the elimination rules only remove states with unsatisfiable labels, so the Δ ‘survives’ the elimination phase and remains in the final tableau. A detailed proof can be found in [2].

Theorem 2. *The tableaux method for* ATL$^+$ *is complete.*

Completeness of the procedure means that an open tableau implies existence of a CGM model. This is proved by first introducing the notion of *Hintikka game structure*, which is essentially a partially defined CGM, and showing that every open tableau provides a Hintikka game structure containing the input formula η in the label of a state in it, which is equivalent of η being satisfiable. Again, a detailed proof can be found in [2].

Theorem 3. *The tableaux procedure for* ATL^+ *runs in* 2EXPTIME.

Proof. The argument generally follows the calculations computing the complexity of the tableaux method for ATL in Section 4.7 of [7], with one essential difference: $\|cl(\eta)\|$ for any ATL formula η is linear in its length $|\eta|$, whereas $\|cl(\eta)\|$ for an ATL^+ formula η can be exponentially large in $|\eta|$, as shown after Lemma 2. This exponential blowup, combined with the worst-case exponential in $\|cl(\eta)\|$ number of states in the tableaux, accounts for the 2EXPTIME worst-case complexity of the tableaux method for ATL^+, which is the expected optimal lower bound. It is also an upper bound for the tableaux method, because no further exponential blowups occur in the prestate- and state-elimination phases. □

There are various ways to restrict or parameterize the set of ATL^+ formulae in order to avoid the exponential blowup of their closure sets. As suggested by the example after Lemma 2, the main cause for that blowup of the number of γ-components of a γ-formulae $\varphi = \langle\langle A\rangle\rangle\Phi$ or $\varphi = [\![A]\!]\Phi$ in ATL^+ is the nesting of conjunctions and disjunctions in the path formula Φ which are not in the scope of temporal operators. Let us call that number the *superficial Boolean depth* of Φ and denote it by $\delta_0(\Phi)$. Then, let the *nested Boolean depth* of any ATL^+ formula Ψ, denoted $\delta(\Psi)$, be the maximal superficial Boolean depth $\delta_0(\Phi)$ of a path subformula Φ of Ψ. For instance, $\delta(\langle\langle 1\rangle\rangle\bigcirc\langle\langle 1\rangle\rangle((p\vee q)\,\mathcal{U}\neg q)) = 0$, $\delta(\langle\langle 1\rangle\rangle(\Box p\vee ((q\wedge p)\,\mathcal{U}\neg q)) = 1$, $\delta(\langle\langle 1\rangle\rangle(\Diamond q\wedge(\Box p\wedge(q\,\mathcal{U}\neg q))) = 2$. Now, if this number for a formula η is bounded above, the size of the closure $\|\eta\|$ becomes polynomially bounded in $|\eta|$ because the nesting of $\wedge$ and $\vee$ when they are separated by a temporal operator does not have multiplicative effect on the number of γ-components. Consequently, the complexity of the tableaux method is reduced to single exponential time, caused only by the maximal possible number of states in the tableaux, just like in ATL. Thus, we have the following.

Proposition 2. *The tableaux procedure for* ATL^+ *applied to a class of* ATL^+ *formulae of bounded nested Boolean depth runs in* EXPTIME.

6 Concluding remarks

Here we have developed sound, complete and terminating tableaux-based decision method for constructive satisfiability testing of ATL^+ formulae and have argued for its practical usability and implementability. The method is amenable to further extension to the full ATL^*, but this is left to future work.

Some comparison with the automata-based method for satisfiability testing in ATL^*, presented in [10] are in order. The two methods appear to be quite different and, though eventually working in the same worst case complexity, the double exponential blowups seem to occur in different ways, namely, in the automata-based method, one exponential blowup occurs in converting the formula into an automaton, while the other is in the time complexity of checking non-emptiness of the resulting automaton. It would be instructive to compare the practical implications and efficiency of both methods and we leave such systematic comparison to the future, when (hopefully) both methods are implemented. For now, we only mention that the formula θ from our running example, the tableau for which is worked out explicitly and in detail in this paper, is translated with the method from [10] into an automaton with 2^{12} alphabet symbols and over 100 states. Of course, this comparison cannot serve as an argument for general practical superiority in efficiency of the tableaux-based method. Still, the technical details of both methods, illustrated in that example, indicate that, while the worst case exponential blowups are bound to occur in both methods, they seem to be more controllable and avoidable in the tableaux-based method, at the expense of its lesser automaticity and higher degree of user control. Thus, we would argue that both methods have generally incomparable pros and cons, and consequently are of independent interest, both theoretically and practically.

Acknowledgments. We thank the anonymous reviewers for their helpful remarks and suggestions and for some corrections. We also gratefully acknowledge the financial support for this research from Université Évry Val-d'Essonne and the laboratory IBISC by funding visits of V. Goranko to Université Évry Val-d'Essonne and of A. David to the Technical University of Denmark.

References

1. Alur, R., Henzinger, T.A., Kuperman, O.: Alternating-time temporal logic. Journal of the ACM 49(5), 672–713 (2002)
2. Cerrito, S., David, A., Goranko, V.: Optimal tableaux-based decision procedure for testing satisfiability in the alternating-time temporal logic ATL$^+$. Tech. rep., Laboratoire IBISC - Université Evry Val-d'Essonne (2014), `https://www.ibisc.univ-evry.fr/~adavid/fichiers/IJCAR_Extended_Draft_Tableau_ATLplus.pdf`
3. David, A.: TATL: Implementation of ATL tableau-based decision procedure. In: Proc. of TABLEAUX'2013, Springer LNCS 8123. pp. 97–103 (2013)
4. van Drimmelen, G.: Satisfiability in alternating-time temporal logic. In: Proc. of LICS'2003. pp. 208–217 (2003)
5. Emerson, E.A.: Temporal and modal logics. In: van Leeuwen, J. (ed.) Handbook of Theoretical Computer Science, vol. B, pp. 995–1072. MIT Press (1990)
6. Goranko, V., van Drimmelen, G.: Complete axiomatization and decidablity of Alternating-time temporal logic. Theor. Comp. Sci. 353, 93–117 (2006)
7. Goranko, V., Shkatov, D.: Tableau-based decision procedures for logics of strategic ability in multiagent systems. ACM Trans. Comput. Log. 11(1) (2009)
8. Johannsen, J., Lange, M.: CTL+ is Complete for Double Exponential Time. In: ICALP'03. LNCS, vol. 2719, pp. 767–775. Springer (2003)
9. Pratt, V.R.: A near optimal method for reasoning about action. Journal of Computer and System Sciences 20, 231–254 (1980)
10. Schewe, S.: ATL* satisfiability is 2EXPTIME-complete. In: Proc. of ICALP (2). pp. 373–385 (2008)
11. Walther, D., Lutz, C., Wolter, F., Wooldridge, M.: ATL satisfiability is indeed ExpTime-complete. Journal of Logic and Computation 16(6), 765–787 (2006)
12. Wolper, P.: The tableau method for temporal logic: an overview. Logique et Analyse 28(110–111), 119–136 (1985)