Optimal decision procedures for satisfiability in fragments of alternating-time temporal logics

Valentin Goranko\textsuperscript{a,b} Steen Vester\textsuperscript{a} \textsuperscript{1}

\textsuperscript{a}Department of Applied Mathematics and Computer Science
Technical University of Denmark

\textsuperscript{b}Department of Mathematics, University of Johannesburg

Abstract

We consider several natural fragments of the alternating-time temporal logics ATL\textsuperscript{*} and ATL with restrictions on the nesting between temporal operators and strategic quantifiers. We develop optimal decision procedures for satisfiability in these fragments, showing that they have much lower complexities than the full languages. In particular, we prove that the satisfiability problem for state formulae in the full 'strategically flat' fragment of ATL\textsuperscript{*} is PSPACE-complete, whereas the satisfiability problems in the flat fragments of ATL and ATL\textsuperscript{+} are $\Sigma^p_3$-complete. We note that the nesting hierarchies for fragments of ATL\textsuperscript{*} collapse in terms of expressiveness above nesting depth 1, hence our results cover all such fragments with lower complexities.

Keywords: satisfiability, decision procedures, alternating-time temporal logics, flat fragments, complexity

1 Introduction

The Alternating-time temporal logic ATL\textsuperscript{*} was introduced and studied in [2] as a multi-agent extension of the branching time temporal logic CTL\textsuperscript{*}, applied for specification and verification of properties of open systems. The most natural semantics for ATL\textsuperscript{*} is defined in multi-agent transition systems, also known as concurrent game models, in which all agents take simultaneous actions at the current state and the resulting collective action effects the state transition. The language of ATL\textsuperscript{*} involves expressions of the type $\langle\langle C\rangle\rangle\Phi$ meaning that the coalition of agents $C$ has a collective strategy to guarantee – no matter how the other agents choose to act – achieving the goal $\Phi$ on all plays (computations) enabled by that collective strategy. The logic ATL\textsuperscript{*} and its fragment ATL (analogous to CTL) have gradually become one of the most popular logical formalisms for reasoning about multi-agent systems, studied extensively during the past 10 years both from a logical and computational perspective.

While found to be quite useful and natural, however, the logic ATL\textsuperscript{*}, and even its fragment ATL, turned out to have some problematic semantic features

\textsuperscript{1} Emails: vfgo@dtu.dk, stve@dtu.dk
related to the nesting of strategic quantifiers $\langle \cdot \rangle$, including:

- Conceptual difficulty in understanding the very meaning of nested expressions of the type $\langle A \rangle \ldots \langle B \rangle \Phi$, especially, when the coalitions $A$ and $B$ share common agents. For instance, what exactly should $\langle A \rangle \neg \langle A \rangle \Phi$ mean?

- This problem is related to a technical problem built in the semantics of $\text{ATL}^*$, where e.g., in the truth evaluation of a formula of the type $\langle A \rangle \ldots \langle B \rangle \Phi$ the strategy for $A$ adopted to guarantee the success of the goal $\ldots \langle B \rangle \Phi$ does not have any effect when evaluating the truth of the subgoal $\langle B \rangle \Phi$, which, arguably, goes against the intuitive understanding of what a strategy and its execution mean. Such problems have lead to several proposals of alternative semantics for $\text{ATL}^*$, with irrevocable commitment to strategies [1] or with strategy contexts, explicitly controllable within the formulae [3]. The latter comes at a high price, resulting in an undecidable satisfiability problem [15].

- The meaning of $\text{ATL}^*$ formulae with nested strategic quantifiers is sensitive to the ability and capacity of the agents to use memory in their strategies, leading to essential variations of the semantics [4].

- The complexity of the full $\text{ATL}^*$ is very high: 2EXPTIME-complete for both the model checking [2] and the satisfiability testing [12] problems.

- These problems are amplified when incomplete information is assumed. Then the basic temporal operators can no longer be naturally (if at all) characterized as fixed points of suitable operators, the semantics becomes truly non-computational and even model checking of $\text{ATL}$ becomes undecidable [2].

So, there are several good reasons to consider flat fragments of $\text{ATL}^*$, where nesting of strategic quantifiers and temporal operators is restricted or completely disallowed, thus avoiding the problems listed above at the cost of reduced expressiveness. There are two natural kinds of ‘flatness’ in the language of $\text{ATL}^*$: with respect to the temporal operators and with respect to strategic quantifiers. The former comes naturally from purely temporal logics and has been investigated before, see e.g., [9], [5], and [13] from a more general, coalgebraic perspective. Here we will mainly consider the latter type of flatness.

The objective of the present paper is to develop optimal algorithmic methods for solving the satisfiability problem for the variety of naturally definable flat fragments of $\text{ATL}^*$ and to analyze their computational complexity. Our main results and the contributions of this paper are as follows:

(i) The algorithmic problem of satisfiability testing in the full fragment of $\text{ATL}^*$ where nesting between strategic quantifiers is not allowed (but temporal operators can be nested in strategic quantifiers and between each other) is PSPACE-complete, in contrast to the 2EXPTIME-completeness of satisfiability in the full $\text{ATL}^*$.

(ii) The algorithmic problem of satisfiability testing in the flat fragments of $\text{ATL}$ and $\text{ATL}^+$, where only nesting of temporal operators in the scope of strategic quantifiers is allowed, are $\Sigma^P_3$-complete, in contrast to the
2EXPTIME-completeness of that problem in the full $\text{ATL}^*$ (as subsuming $\text{CTL}^*$, see [10]) and its EXPTIME-completeness in the full $\text{ATL}$ [8].

The structure of the paper is as follows: In Section 2 we summarize basics of the logics $\text{LTL}$, $\text{CTL}$ and $\text{CTL}^*$ as well as concurrent game models and the alternating-time temporal logics $\text{ATL}$, $\text{ATL}^+$ and $\text{ATL}^*$. In Section 3 we introduce various flat fragments of $\text{ATL}^*$ and discuss their expressiveness. Section 4 contains the technical preparation for our algorithms, where we introduce some kinds of normal forms for $\text{ATL}^*$ formulae and obtain some key technical results. In Section 5 we provide sound and complete decision procedures as well as matching lower bounds for the flat fragments of $\text{ATL}^*$ considered in the paper. We end with brief concluding remarks in Section 6.

2 Preliminaries

2.1 Summary of $\text{LTL}$, $\text{CTL}$, $\text{CTL}^*$ and their flat fragments

We assume that the reader is familiar with the temporal logics $\text{LTL}$, $\text{CTL}$ and $\text{CTL}^*$. A standard reference is e.g., [6].

Given a set of atomic propositions $\text{Prop}$, the set of literals over $\text{Prop}$ is $\text{Prop} \cup \{\neg p \mid p \in \text{Prop}\}$. We assume that the primitive temporal operators in $\text{LTL}$ and $\text{CTL}^*$ are $\text{X}$ (“at the next state”) and $\text{U}$ (“Until”), whereas $\text{F}$ (“sometime in the future”), $\text{R}$ (“Release”), and $\text{G}$ (“always in the future”) are definable as follows: $\text{F} \varphi := \top \text{U} \varphi$, $\text{ψ R} \varphi := \neg((\neg \varphi) \text{U} (\neg \varphi))$, $\text{G} \varphi := \bot \varphi$. Respectively, the primitive temporal operators in $\text{CTL}$ are $\text{AX}$, $\text{A U}$ and $\text{A R}$, whereas the rest are definable as follows: $\text{EX} \varphi := \neg \text{AX} \neg \varphi$, $\text{E(ψ U} \varphi) := \neg \text{A(} (\neg \varphi) \text{R (} \neg \varphi)\text{)}$, $\text{E} (\psi \text{R} \varphi) := \neg \text{A} (\neg \text{U} (\neg \varphi))$, $\text{AF} \varphi := \text{A(U} \varphi\text{)}$, $\text{AG} \varphi := \text{A(U R} \varphi\text{)}$, $\text{EF} \varphi := \text{E(U} \varphi\text{)}$, $\text{EG} \varphi := \text{E(U R} \varphi\text{)}$.

The following $\text{LTL}$-equivalences characterize $\text{U}$ and $\text{R}$ as fixed points, where the formulae on the right hand side are called the fixed point unfoldings respectively of $\theta \text{U} \eta$ and $\theta \text{R} \eta$ (see e.g., [14], [6]):

\[\theta \text{U} \eta \equiv \eta \lor (\theta \land \text{X} (\theta \text{U} \eta)), \quad \theta \text{R} \eta \equiv \eta \land (\theta \lor \text{X} (\theta \text{R} \eta)).\]

We define the flat fragments $\text{LTL}_1$, $\text{CTL}_1$ and $\text{CTL}_1^*$ resp. as subsets of $\text{LTL}$, $\text{CTL}$ and $\text{CTL}^*$. In $\text{LTL}_1$ no nesting of temporal operators is allowed, in $\text{CTL}_1^*$ no nesting of path quantifiers is allowed and in $\text{CTL}_1$ neither is allowed. They are generating as follows, where $\beta$ is a Boolean formula and $\theta$ is an $\text{LTL}$ formula:

- $\text{LTL}_1$: $\theta ::= p \mid \neg \theta \mid \theta \land \theta \mid \text{X} \beta \mid \beta \text{U} \beta$;
- $\text{CTL}_1^*$: $\varphi ::= p \mid \neg \varphi \mid \varphi \land \varphi \mid \text{AX} \beta \mid (\beta \text{U} \beta) \mid (\beta \text{R} \beta)$.

For instance:

- $p \text{U} q \land \text{X} (r \land (q \land \neg p))$ is in $\text{LTL}_1$ but $p \text{U} (\text{X} q)$ is not.
- $\text{A(} \neg \text{p U(} p \land \neg q\text{)}\text{)} \land (\text{EF (} q \land \neg p\text{)} \land \neg \text{A F(} \neg(p \land q)\text{)})$ is in $\text{CTL}_1$ (and in $\text{CTL}_1^*$).
- $\text{AG F (} p\text{)}$ is in $\text{CTL}_1^*$ but not in $\text{CTL}_1$; $\text{AG EF (} p\text{)}$ is neither in $\text{CTL}_1$ nor in $\text{CTL}_1^*$.
2.2 Concurrent game models. The logic ATL* and fragments

A concurrent game model [2] (CGM) is a tuple \( M = (A, St, \{Act_a\}_{a \in A}, \{act_a\}_{a \in A}, out, Prop, L) \) comprising:

- a finite, non-empty set of players (agents) \( A = \{1, \ldots, k\} \)
- a set of actions \( Act_a \neq \emptyset \) for each \( a \in A \). For any \( A \subseteq \hat{A} \) we denote \( Act_A := \bigcap_{a \in A} Act_a \) and use \( \alpha_A \) to denote a tuple from \( Act_A \). In particular, \( Act_{\hat{A}} \) is the set of all possible action profiles in \( M \).
- a non-empty set of states \( St \),
- for each \( a \in A \), a map \( act_a : St \to \mathcal{P}(Act_a) \setminus \{\emptyset\} \) setting for each state \( s \) the actions available to \( a \) at \( s \),
- a transition function \( out : St \times Act_{\hat{A}} \to St \) that assigns deterministically a successor (outcome) state \( out(s, \alpha_{\hat{A}}) \) to every state \( s \) and action profile \( \alpha_{\hat{A}} = \langle \alpha_1, \ldots, \alpha_k \rangle \), provided that \( \alpha_s \in act_a(s) \) for every \( a \in \hat{A} \) (i.e., every \( \alpha_a \) that can be executed by player \( a \) in state \( s \)),
- a finite set of atomic propositions \( Prop \) and a labelling \( L : St \to \mathcal{P}(Prop) \).

Concurrent game models represent multi-agent transition systems that function as follows: at any moment the system is in a given state, where each player select an action from those available to him at that state. All players 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 such subsequent successor states. More formally, a play is an infinite sequence \( s_0 s_1 \ldots \in St^\omega \) of states such that for each \( i \geq 0 \) there exists an action profile \( \alpha_{\hat{A}} = \langle \alpha_1, \ldots, \alpha_k \rangle \) such that \( out(s_i, \alpha_{\hat{A}}) = s_{i+1} \). A history is a finite initial segment \( s_0 s_1 \ldots s_{\ell} \) of a play. We denote by \( \text{Play}_M \) and \( \text{Hist}_M \) respectively the set of plays and set of histories in \( M \). For a state \( s \in St \) we define \( \text{Play}_M(s) \) and \( \text{Hist}_M(s) \) as the set of plays and set of histories with initial state \( s \). For a sequence \( \rho \) of states \( \rho_0 \) is the initial state, \( \rho_i \) is the \( (i+1) \)th state, \( \rho_{\leq i} \) is the prefix \( \rho_0 \rho_1 \ldots \rho_i \) of \( \rho \) and \( \rho_{> i} \) is the suffix \( \rho_{i+1} \rho_{i+2} \ldots \) of \( \rho \). When \( \rho = \rho_0 \ldots \rho_{\ell} \) is finite, we say that it has length \( \ell \) and write \( |\rho| = \ell \). Further, we let \( \text{last}(\rho) = \rho_{\ell} \).

A strategy for a player \( a \) in \( M \) is a mapping \( \sigma_a : \text{Hist}_M \to Act_a \) such that for all \( h \in \text{Hist}_M \) we have \( \sigma_a(h) \in act_a(\text{last}(h)) \). Intuitively, it assigns a legal action for player \( a \) after any history \( h \) of the game. If that action depends only on the current state, the strategy is called memoryless. We denote by \( \text{Strat}_M(a) \) the set of strategies of player \( a \). A (collective) strategy of a coalition \( C \subseteq \hat{A} \) is a tuple \( \langle \sigma_a \rangle_{a \in C} \) of strategies, one for each player in \( C \). When \( C = \hat{A} \) this is called a strategy profile. We denote by \( \text{Strat}_M(C) \) the set of collective strategies of coalition \( C \). A play \( \rho \in \text{Play}_M \) is consistent with a strategy \( \sigma_C \in \text{Strat}_M(C) \) if for every \( i \geq 0 \) there exists an action profile \( \alpha_{\hat{A}} = \langle \alpha_1, \ldots, \alpha_k \rangle \) such that \( out(\rho_i, \alpha_{\hat{A}}) = \rho_{i+1} \) and \( \sigma_a(\rho_{\leq i}) \) for all \( a \in C \). The set of plays with initial state \( s \) that are consistent with \( \sigma_C \) is denoted \( \text{Play}_M(s, \sigma_C) \). In particular, we define \( \text{Play}_M(s, \sigma_a) = \text{Play}_M(s, \sigma_{\{a\}}) \) for any player \( a \).

The Alternating-time temporal logic ATL*, introduced in [2], is a logic, suitable for specifying and verifying qualitative objectives of players and coali-
tions in concurrent game models. The main syntactic construct of \( \text{ATL}^+ \) is a formula of type \( ⟨⟨C⟩⟩Φ \), intuitively meaning: “The coalition \( C \) has a collective strategy to guarantee the satisfaction of the objective \( Φ \) on every play enabled by that strategy.” Formally, \( \text{ATL}^+ \) is a multi-agent extension of the branching time logic CTL* with strategic quantifiers \( ⟨⟨C⟩⟩ \) indexed with sets (coalitions) \( C \) of players. There are two types of formulae in \( \text{ATL}^+ \), \text{state formulae}, that are evaluated at states, and \text{path formulae}, that are evaluated on plays. These are defined by mutual recursion as follows, where \( C \subseteq A \), \( p \in \text{Prop} \):

\[
\text{State formulae of } \text{ATL}^+ : \varphi := p \mid ¬\varphi \mid \varphi \land \varphi \mid ⟨⟨C⟩⟩Φ.
\]

\[
\text{Path formulae of } \text{ATL}^+ : Φ := ¬Φ \mid Φ \land Φ \mid X\varphi \mid Φ \lor Φ \mid Φ \land Φ \mid Φ R Φ.
\]

All other Boolean connectives are defined as usual, and the temporal operators \( F \) and \( G \) are defined as in CTL*, which can be regarded as the fragment of \( \text{ATL}^+ \) only involving strategic quantifiers for the empty coalition \( ⟨⟨∅⟩⟩ \), identified with universal path quantifier \( A \), and for the “grand coalition” of all players \( ⟨⟨A⟩⟩ \), identified with existential path quantifier \( E \). Equivalently, by identifying all agents, CTL* can be regarded as the 1-agent fragment of \( \text{ATL}^+ \). To keep the notation lighter, we will list the members of \( C \) in \( ⟨⟨C⟩⟩ \) without using \( \{\} \).

The fragment \( \text{ATL}^+ \) of \( \text{ATL}^+ \) is obtained when the temporal operators may only be applied to state formulae, i.e. when path formulae are re-defined as

\[
\text{Path formulae of } \text{ATL}^+ : Φ := ¬Φ \mid Φ \land Φ \mid X\varphi \mid Φ \lor Φ \mid Φ R Φ.
\]

Another, technically simpler and computationally better behaved fragment of \( \text{ATL}^+ \), is the logic ATL, which is the multi-agent analogue of CTL, only involving state formulae defined as follows, for any \( C \subseteq A \), \( p \in \text{Prop} \):

\[
\text{Formulæ of } \text{ATL} : \varphi := p \mid ¬\varphi \mid \varphi \land \varphi \mid ⟨⟨C⟩⟩X\varphi \mid ⟨⟨C⟩⟩(ϕ \lor ϕ) \mid ⟨⟨C⟩⟩(ϕ Rϕ).
\]

The combined operators \( ⟨⟨C⟩⟩Fϕ \) and \( ⟨⟨C⟩⟩Gϕ \) are defined respectively as \( ⟨⟨C⟩⟩Tϕ \) and \( ⟨⟨C⟩⟩Lϕ \).

The semantics of \( \text{ATL}^+ \) is given with respect to a concurrent game model \( M = (A, St, \{\text{Act}_a\}_{a \in A}, \{act_s\}_{s \in \text{Prop} \lor \text{L}}, \text{out}, \text{Prop}, \text{L}) \). The semantics of state formulae is given in terms of truth at a state \( s \) in \( M \), as follows, where \( p \in \text{Prop} \), \( ϕ_1 \) and \( ϕ_2 \) are state formulae, \( Φ \) is a path formula and \( C \subseteq A \):

\[
\mathcal{M}, s \models p \quad \text{if } p \in \text{L}(s).
\]

\[
\mathcal{M}, s \models ¬ϕ_1 \quad \text{if } \mathcal{M}, s \not\models ϕ_1.
\]

\[
\mathcal{M}, s \models ϕ_1 \land ϕ_2 \quad \text{if } \mathcal{M}, s \models ϕ_1 \text{ and } \mathcal{M}, s \models ϕ_2.
\]

\[
\mathcal{M}, s \models ⟨⟨C⟩⟩Φ \quad \text{if there exist a collective strategy } σ_C ∈ \text{Strat}_M(C), \text{ such that } \mathcal{M}, ρ \models Φ \text{ for all } ρ ∈ \text{Play}_M(s, σ_C).
\]

The semantics of path formulae is given just like in LTL, in terms of truth on a path \( ρ \) in a CGM \( M \), as follows, where \( ϕ \) is a state formula, \( Φ_1 \) and \( Φ_2 \) are path formulae and \( C \subseteq A \):

\[
\mathcal{M}, ρ \models Φ_1 \quad \text{if } ϕ \text{ holds on } ρ \text{ and } Φ_1 \text{ holds on } ρ.
\]

\[
\mathcal{M}, ρ \models Φ_2 \quad \text{if } ϕ \text{ holds on } ρ \text{ and } Φ_2 \text{ holds on } ρ.
\]

\[
\mathcal{M}, ρ \models Φ \quad \text{if } ϕ \text{ holds on } ρ \text{ and } Φ \text{ holds on } ρ.
\]
\[ M, \rho \models \varphi \quad \text{if} \quad M, \rho_0 \models \varphi \]
\[ M, \rho \models \neg \Phi_1 \quad \text{if} \quad M, \rho \not\models \Phi_1 \]
\[ M, \rho \models \Phi_1 \land \Phi_2 \quad \text{if} \quad M, \rho \models \Phi_1 \text{ and } M, \rho \models \Phi_2 \]
\[ M, \rho \models X \Phi_1 \quad \text{if} \quad M, \rho_0 \models X \Phi_1 \]
\[ M, \rho \models \Phi_1 \text{ and } M, \rho \models \Phi_2 \quad \text{if} \quad \exists k. M, \rho_k \models \Phi_1 \text{ and } \forall j < k. M, \rho_j \models \Phi_2 \]
\[ M, \rho \models \Phi_1 \quad \text{if} \quad \forall k. M, \rho_k \models \Phi_1 \text{ or } \exists k. M, \rho_k \models \Phi_1 \text{ and } \forall j < k. M, \rho_j \models \Phi_1 \]

We focus on the satisfiability problem for various fragments of ATL* in this paper. We will distinguish between the state satisfiability and path satisfiability problems which are defined on a given fragment \( \mathcal{L} \) of ATL* as follows:

- Given a state formula \( \varphi \) in \( \mathcal{L} \), does there exist a CGM \( M \) and a state \( s \) in \( M \) such that \( M, s \models \varphi \)?
- Given a path formula \( \Phi \) in \( \mathcal{L} \), does there exist a CGM \( M \) and a play \( \rho \) in \( M \) such that \( M, \rho \models \Phi \)?

Note that there are two variants of the satisfiability problem for formulae of ATL*: 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. It is easy to see that 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 \( \langle a \rangle X \top \), so we hereafter only consider the tight satisfiability version. For further details and discussion on this issue, see e.g., [7,17].

We recall some important complexity results for the satisfiability problem: satisfiability in ATL is EXPTIME-complete [16,8], while satisfiability in ATL* is 2EXPTIME-complete [12]. Since ATL* subsumes CTL+, the satisfiability in which is also 2EXPTIME-complete [10], this is the optimal complexity for the satisfiability in ATL*, too. All these results equally hold for satisfiability in concurrent game models and in alternating transition systems [2], as both semantics are equivalent (see e.g., [8]).

3 Flat fragments of ATL and ATL*

Here we define some flat fragments of ATL* and ATL. Flatness generally means no nesting of non-Boolean operators. There are two natural notions of flatness in the languages of ATL and ATL*:

- with respect to temporal operators and
- with respect to strategic quantifiers. We will be mostly concerned with the latter, but the former applies in the case of ATL, too.

We adopt the following notational conventions: we will typically denote Boolean formulae by \( \beta, \gamma \); LTL formulae by \( \theta, \eta, \zeta \); ATL formulae by \( \varphi, \psi \); and ATL* formulae – both state and path – by \( \Theta, \Phi, \Psi \); all possibly with indices.

3.1 A hierarchy of flat fragments of ATL*

We will consider the following fragments of ATL*, where \( p \) is any atomic proposition, \( C \subseteq A \), \( \beta \) is any Boolean formula and \( \theta \) is any LTL formula:
(i) **Separated** \(\text{ATL}^*\), denoted \(\text{ATL}^\text{sep}_*\), consists of those formulae of \(\text{ATL}^*\) in which there is no nesting of strategic quantifiers in the scope of temporal operators (but, any nesting of temporal operators within strategic quantifiers or temporal operators is allowed), so the (external) strategic and the (internal) temporal layers are separated. More precisely, the formulae of \(\text{ATL}^\text{sep}_*\) are generated as follows:

\[
\Phi ::= \theta \mid \neg \Phi \mid (\Phi \land \Phi) \mid (\langle\langle C \rangle\rangle \Phi)
\]

(ii) **Full (strategically) flat** \(\text{ATL}^*\), denoted \(\text{ATL}^*_1\), consists of those formulae of \(\text{ATL}^*\) in which there is no nesting of strategic quantifiers within strategic quantifiers (but, nesting of strategic quantifiers and temporal operators in temporal operators is allowed), formally generated as follows:

\[
\Phi ::= p \mid \neg \Phi \mid (\Phi \land \Phi) \mid (\langle\langle C \rangle\rangle \theta) \mid X \Phi \mid \Phi U \Phi \mid \Phi R \Phi
\]

If the restriction \(C \neq \emptyset\) is imposed, we denote the resulting fragment \(\hat{\text{ATL}}^*_1\).

(iii) **State fragment** of \(\text{ATL}^*_1\), denoted \(\text{St}(\text{ATL}^*_1)\), consists of the state formulae of \(\text{ATL}^*_1\), i.e. those formulae of \(\text{ATL}^*\) in which there is no nesting of strategic quantifiers in either temporal operators or strategic quantifiers (but, nesting between temporal operators is allowed). The formulae of \(\text{St}(\text{ATL}^*_1)\) are explicitly generated as follows:

\[
\Phi ::= p \mid \neg \Phi \mid (\Phi \land \Phi) \mid (\langle\langle C \rangle\rangle \theta).
\]

(iv) **Flat** \(\text{ATL}^+_1\) (or, **double-flat** \(\text{ATL}^*\), denoted \(\text{ATL}^+_1\), consists of those formulae of \(\text{ATL}^+\) which are also in \(\text{St}(\text{ATL}^*_1)\), e.g., with no nesting of either strategic quantifiers or temporal operators within temporal operators. The formulae of \(\text{ATL}^+_1\) are generated as follows, where \(\theta \in \text{LTL}_1\):

\[
\Phi ::= p \mid \neg \Phi \mid (\Phi \land \Phi) \mid (\langle\langle C \rangle\rangle \theta).
\]

(v) **Flat** \(\text{ATL}_1\), denoted \(\text{ATL}_1\), consists of those formulae of \(\text{ATL}^+_1\) which are in \(\text{ATL}\), i.e., in which strategic quantifiers are followed immediately by temporal operators. The formulae of \(\text{ATL}_1\) are generated as follows:

\[
\varphi ::= p \mid \neg \varphi \mid (\varphi \land \varphi) \mid (\langle\langle C \rangle\rangle X \beta) \mid (\langle\langle C \rangle\rangle (\beta U \beta)) \mid (\langle\langle C \rangle\rangle (\beta R \beta))
\]

Inclusions between the different flat fragments are illustrated in Figure 1. All inclusions shown in the figure are strict and there are no inclusions except the ones shown (where transitive closure is implicit). For example:

- \(\langle\langle 1\rangle\rangle G \neg p \land \neg \langle\langle 2\rangle\rangle X (p \lor \neg q) \lor \langle\langle 1, 2\rangle\rangle (p U \neg q)\) is in \(\text{ATL}_1\);
- \(\langle\langle 1\rangle\rangle (G \neg p \land F q)\), \(\langle\langle 1, 2\rangle\rangle ((p R \neg q) \lor (\neg p U q))\) are in \(\text{ATL}^+_1\) but not in \(\text{ATL}_1\);
- \(\langle\langle 1\rangle\rangle (GF \neg p \lor X (\neg (p U \neg q)))\) is in \(\text{St}(\text{ATL}^*_1)\) but not \(\text{ATL}^+_1\);
- \(G \langle\langle 1, 2\rangle\rangle F (\neg p \lor q)\) is in \(\hat{\text{ATL}}^*_1\) but not in \(\text{St}(\text{ATL}^*_1)\);
Optimal decision procedures for satisfiability in fragments of alternating-time temporal logics

Fig. 1. Inclusions between flat fragments. An arrow from $L_1$ to $L_2$ means that every $L_1$ formula is an $L_2$ formula.

- $\langle\langle\emptyset\rangle\rangle Gp \land G\langle\langle1,2\rangle\rangle GF\neg p$ is in $\text{ATL}_1^{+}$ but not in $\text{ATL}_1^{\ast}$
- $\langle\langle2\rangle\rangle\langle\langle1\rangle\rangle\neg (p U \neg q)$ is in $\text{ATL}_1^{\ast}\text{Sep}$ but not in $\text{ATL}_1^{+}$

Even though $\text{ATL}_1^{+}$ is included in $\text{ATL}_1^{\ast}\text{Sep}$ they have the same expressive power and there is an efficient translation from $\text{ATL}_1^{\ast}\text{Sep}$ to $\text{ATL}_1^{+}$.

Proposition 3.1 Every formula of $\text{ATL}_1^{\ast}\text{Sep}$ is logically equivalent to a formula of $\text{ATL}_1^{\ast}$ which is at most as long and has no nesting of strategic quantifiers. Such a formula is effectively computable in linear time.

Proof. Because $\langle\langle C\rangle\rangle \Phi \equiv \Phi$ for every state formula $\Phi$ and coalition $C$. □

Thus, deciding satisfiability in $\text{ATL}_1^{\ast}\text{Sep}$ is reducible with no cost to satisfiability in $\text{St}(\text{ATL}_1^{+})$, so we will not discuss $\text{ATL}_1^{\ast}\text{Sep}$ hereafter. On the other hand, due to the equivalence above, the fragment $\text{ATL}_1^{+}$ can be extended even further by allowing nesting of strategic quantifiers, as long as there are no occurrences of temporal operators in between them. The equivalence of $\langle\langle C\rangle\rangle \Phi$ and $\Phi$ for state formulae $\Phi$ is an example of why nesting of strategic quantifiers in $\text{ATL}_1^{\ast}$ can be considered unnatural. We note that this phenomenon is avoided in $\text{ATL}_1^{\ast}$ with strategy context [3].

Between the full logics and their flat fragments, it is natural to consider the hierarchies of fragments with a bounded nesting depth of strategic quantifiers. However, the next result shows that the fragments with nesting depth 2 are essentially as expressive and computationally hard as the full logics.

Proposition 3.2 For any logic $L \in \{\text{LTL}, \text{CTL}, \text{CTL}^{+}, \text{CTL}^{\ast}, \text{ATL}, \text{ATL}^{+}, \text{ATL}^{\ast}\}$ and formula $\Phi$ of $L$ there is an equi-satisfiable formula $\Phi'$ in $L$ with nesting depth 2 of strategic quantifiers (resp., temporal operators for $\text{LTL}$) and length $|\Phi'| = O(|\Phi|)$ that can be computed in linear time.

Proof. In each of the cases, the flattening is done by repeated renaming of state subformulae with fresh atomic propositions. We illustrate the technique on $\text{ATL}^{\ast}$. Let $\Phi$ be an $\text{ATL}^{\ast}$ formula. For any innermost subformula $\Psi$ of $\Phi$ beginning with a strategic quantifier we introduce a fresh atomic proposition $p_{\Psi}$. Then $\Phi$ and $\Phi' = \Phi[p_{\Psi}/\Psi] \land \text{AG}(p_{\Psi} \leftrightarrow \Psi)$ are equi-satisfiable. By repeated application of such renaming of strategically quantified subformulae we obtain an equi-satisfiable formula of nesting depth 2 that is linear in
the size of $\Phi$. Since $AG(p \leftrightarrow \Psi)$ is a CTL formula, this works for each logic $L \in \{\text{CTL}, \text{CTL}^+, \text{CTL}^*, \text{ATL}, \text{ATL}^+, \text{ATL}^*\}$, while for LTL we use $G(p \leftrightarrow \Psi)$.

Thus, the only complexity gain in restricting syntactic fragments of these logics with respect to nesting can occur when flat fragments are considered.

### 3.2 Some remarks on the expressiveness of the flat $\text{ATL}^*$-fragments

The lower complexity of the satisfiability in the flat fragments of $\text{ATL}^*$ comes with a price, namely that various properties that require nesting of strategic quantifiers cannot be expressed anymore. However, many interesting and important properties of systems are still expressible. For instance:

- $\langle\langle \text{ctrl} \rangle\rangle G \neg\text{break}$ in $\text{ATL}_1$ specifies that a controller can make sure the system does not break no matter how the environment behaves,
- $\bigwedge_{i=1}^n \langle\langle \text{proc}_i \rangle\rangle GF \text{db}_i \text{access}$, expresses that each process can ensure database access infinitely often,
- $\langle\langle A \rangle\rangle (\theta_{\text{fair}} \rightarrow \theta)$ means that coalition $A$ can make sure that the LTL property $\theta$ is satisfied on all fair paths (where fairness is defined by LTL formula $\theta_{\text{fair}}$).

The semantics of $\text{ATL}^*$ is based on unbounded memory strategies, but it can be restricted and parameterized with the amount of memory that the proponent agents’ strategies can use. The extreme case is the memoryless semantics, where the proponents may only use memoryless strategies. It turns out that satisfiability in $\text{ATL}$, is unaffected by such restrictions, but differences occur in $\text{ATL}^*$ and even in $\text{ATL}^+$. For discussion on these see e.g., [4]. In contrast, using our satisfiability decision procedures developed in Section 5, we will show that all semantics based on different memory restrictions yield the same satisfiable (resp., the same valid) formulae in the flat fragment $\text{ATL}_1^*$.

### 4 Normal forms and satisfiability of special sets in $\text{ATL}^*$

#### 4.1 Negation normal form of $\text{ATL}^*$ formulae

**Definition 4.1** An $\text{ATL}^*$ formula $\Phi$ is in a negation normal form (NNF) if negations in $\Phi$ may only occur immediately in front of atomic propositions.

We now define the dual $\llbracket \cdot \rrbracket$ to the strategic quantifier $\langle\langle \cdot \rangle\rangle$ as usual:

$\llbracket C \rrbracket := \neg\langle\langle C \rangle\rangle \neg$. If we consider $\llbracket \cdot \rrbracket$ as a primitive operator in $\text{ATL}^*$, then every $\text{ATL}^*$ formula can be transformed to an equivalent formula in NNF by driving all negations inwards, using the self-duality of $X$ and the duality between $U$ and $R$. However, using $\llbracket \cdot \rrbracket$ formally breaks the syntax of the fragments $\text{ATL}$ and $\text{ATL}_1$ because of inserting a $\neg$ between $\langle\langle \cdot \rangle\rangle$ and the temporal operator. Yet, this can be easily fixed by equivalently re-defining the applications of $\llbracket \cdot \rrbracket$, using the following equivalences: $\llbracket C \rrbracket X \varphi \equiv \neg\langle\langle C \rangle\rangle X \neg\varphi$, $\llbracket C \rrbracket (\varphi U \psi) \equiv \neg\langle\langle C \rangle\rangle ((\neg\varphi) R (\neg\psi))$, $\llbracket C \rrbracket (\varphi R \psi) \equiv \neg\langle\langle C \rangle\rangle ((\neg\varphi) U (\neg\psi))$.

Hereafter we assume that the language $\text{ATL}^*$ and each of its fragments introduced above are formally extended with the operator $\llbracket \cdot \rrbracket$ applied just like $\langle\langle \cdot \rangle\rangle$ in the respective fragments. Due to the equivalences above, the resulting extensions preserve the expressiveness of these fragments. Formally:
Lemma 4.2. Every formula of ATL* extended with the operator $[[·]]$ can be transformed to an equivalent formula in NNF. Furthermore, each of the fragments ATL, ATL1, ATL1*, St(ATL1) and ATL1*, extended with $[[·]]$, is closed under this transformation, i.e. if a formula is in any of these fragments then its NNF-equivalent formula is in that fragment, too.

4.2 Successor normal forms

Definition 4.3. [Successor formulae] An ATL* formula is a successor formula (SF) if it is of the type $[[C]]X\Phi$ or $[[C]]\Phi$.

Definition 4.4. [Components] With every set of ATL* successor formulae

$$\Gamma = \{\langle A_0 \rangle X\Phi_0, \ldots, \langle A_{m-1} \rangle X\Phi_{m-1}, [[B_0]]X\Psi_0, \ldots, [[B_{n-1}]]X\Psi_{n-1} \}$$

we associate the set of its

- $\langle \cdot \rangle X$-components: $\langle \cdot \rangle X(\Gamma) = \{\Phi_0, \ldots, \Phi_{m-1}\}$,
- $[[\cdot]]X$-components: $[[\cdot]]X(\Gamma) = \{\Psi_0, \ldots, \Psi_{n-1}\}$,
- successor components: $SC(\Gamma) = \langle \cdot \rangle X(\Gamma) \cup [[\cdot]]X(\Gamma)$.

Definition 4.5. [Successor normal form]

(i) An LTL formula is in a LTL successor normal form (LSNF) if it is in NNF and is a Boolean combination of literals and successor formulae, i.e., LTL formulae beginning with $X$.

(ii) An ATL* formula is in a successor normal form (SNF) if it is in NNF and is a Boolean combination of literals and ATL* successor formulae.

Lemma 4.6. Every LTL-formula $\zeta$ can be effectively transformed to an equivalent formula in LTL successor normal form $LSNF(\zeta)$, of length at most $6|\zeta|$.

Proof. We can assume that $\zeta$ is already transformed to NNF (of length less than twice the original length). Consider all maximal subformulae of $\zeta$ of the types $\theta U \eta$ and $\theta R \eta$. Replace each of them with its LTL-equivalent fixpoint unfolding, respectively $\eta \lor (\theta \land X(\theta U \eta))$ and $\eta \land (\theta \lor X(\theta R \eta))$. Then, the same procedure is applied recursively to all respective subformulae $\theta$, $\eta$ occurring above and not in the scope of $X$, until all occurrences of $U$ and $R$ get in the scope of $X$. This procedure at most triples the length of the starting formula and the result is clearly a formula in LSNF.

Definition 4.7. [Conjunctive formulae in SNF] An ATL* formula in SNF is conjunctive if it is of the form

$$\Theta = \Phi \land \langle A_0 \rangle X\Phi_0 \land \ldots \land \langle A_{m-1} \rangle X\Phi_{m-1} \land [[B_0]]X\Psi_0 \land \ldots \land [[B_{n-1}]]X\Psi_{n-1}$$

With every such formula $\Theta$ we associate the set of its successor conjuncts:

$$SC(\Theta) = \{\langle A_0 \rangle X\Phi_0, \ldots, \langle A_{m-1} \rangle X\Phi_{m-1}, [[B_0]]X\Psi_0, \ldots, [[B_{n-1}]]X\Psi_{n-1}\}$$

4.3 Sets of distributed control of ATL* formulae

Definition 4.8. [Set of distributed control] A set of ATL* formulae $\Delta$ is a set of distributed control if $\Delta = \{\langle A_0 \rangle \Phi_0, \ldots, \langle A_{m-1} \rangle \Phi_{m-1}, [[B]]\Psi\}$ where the
coalsitions $A_0, \ldots, A_{l-1}$ are pairwise disjoint, and $A_0 \cup \ldots \cup A_{l-1} \subseteq B$.

**Lemma 4.9** A set of ATL* successor formulae

$$\Gamma = \{ \langle A_0 \rangle X \Phi_0, \ldots, \langle A_{m-1} \rangle X \Phi_{m-1}, [B_0] X \Psi_0, \ldots, [B_{n-1}] X \Psi_{n-1}, [[A]] X \top \}$$

is satisfiable if and only if every subset of distributed control $\Delta$ of $\Gamma$ has a satisfiable set of successor components.

**Proof.** First, note that the formula $[[A]] X \top$ is valid, so it plays no role in the satisfiability of $\Gamma$; it is only added there in order to enable sufficiently many subsets of distributed control.

Now, suppose $\Gamma$ is true at a state $s$ of a CGM $M$. Then for every subset of distributed control $\Delta = \{ \langle A_0 \rangle X \Phi_0, \ldots, \langle A_{l-1} \rangle X \Phi_{l-1}, [B] X \Psi \}$ consider collective actions for the coalitions $A_0, \ldots, A_{l-1}$ at $s$ that guarantee satisfaction of their respective nexttime objectives in $\Delta$ in any of the resulting successor states. Add arbitrarily fixed actions of the remaining agents in $B$ and a respective collective action for $A \setminus B$ dependent on the so fixed actions of the agents in $B$, that brings about satisfaction of $\Psi$ in the resulting successor state $s'$. Then all successor components of $\Delta$ are true at $s'$.

Conversely, suppose that $\Delta_1, \ldots, \Delta_d$ are all subsets of $\Gamma$ of distributed control and they are all satisfiable. For each $\Delta_i$ we fix a CGM $M_i$ and a state $s_i$ in it that satisfies $SC(\Delta_i)$. We can assume, w.l.o.g., that $M_i$ is generated from $s_i$, i.e. consists only of states reachable by plays starting at $s_i$.

We will construct a CGM satisfying $\Gamma$ by using a construction from [8]. The idea is to first create a root state $s$ and supply all agents with sufficiently many actions at $s$ in order to ensure the existence of all collective actions and respective successor states necessary for satisfying the successor components of $\Gamma$. We will show that it suffices to take care of the sets of successor components of each subset of distributed control $\Gamma$ and then will use the CGMs satisfying these to complete the construction of the model satisfying $\Gamma$.

Now, the construction. Recall that $|A_i| = k$ and let $r = m + n$ (the numbers of $\langle \cdot \rangle$- and $[\cdot]$-components in $\Gamma$). Each agent will have $r$ available actions $\{0, \ldots, r - 1\}$ at the root state $s$, hence $\{0, \ldots, r - 1\}^k$ is the set of all possible action profiles at $s$. The intuition is that every agent’s action at $s$ is a choice of that agent of a formula from $\Gamma$ for the satisfaction of which the agent chooses to act. For every such action profile $\sigma$ we denote by $N(\sigma)$ the set of agents $\{ i \mid \sigma_i \geq m \}$ and then we define the number $\text{neg}(\sigma)$ to be the remainder of $\sum_{i \in N(\sigma)} (\sigma_i - m)$ modulo $n$. (The idea of this definition is that, once all agents in any given proper subset of $N(\sigma)$ choose their actions, the remaining agents in $N(\sigma)$ can act accordingly to yield any value of $\text{neg}(\sigma)$ between $0$ and $n - 1$ they wish, i.e., to set the "collective action" of all agents in $\text{neg}(\sigma)$ on any $[[\cdot]] X$-formula in $\Gamma$ they choose.) Now, we consider the set

$$\Delta_\sigma = \{ \langle A_j \rangle X \Phi_j \mid j < m \text{ and } \sigma_j = j \text{ for all } i \in A_j \} \cup \{ [[B_i]] X \Psi_l \mid \text{neg}(\sigma) = l \text{ and } A \setminus B_i \subseteq N(\sigma) \}$$

Note that $\Delta_\sigma$ is a subset of $\Gamma$ of distributed control if it contains a formula
([[B_i]])\times Ψ_i or else can be made a set of distributed control by adding \([[A]]\times T\) to it. Indeed, all agents in a A_j choose j, so all coalitions A_j must be pairwise disjoint. Besides, if \([[B_i]])\times Ψ_i ∈ Δ_σ then it is clearly a unique \([[\_]]\)-formula in Δ_σ and no agents from any A_j ∈ Δ_σ are in N(σ), hence A_j ⊆ B_i for each \(\langle A_j\rangle \times Φ_j \in Δ_σ\). Thus, Δ_σ is one of Δ_1, ..., Δ_d, say Δ_i. Then, we determine the successor state out(s, σ) to be s_i. To complete the definition of the CGM, at each successor state s_i of s we graft a copy of M_i.

We will show that the resulting CGM M satisfies Γ at s. Indeed, for every \(\langle A_j\rangle \times Φ_j \in Γ\) a collective strategy for A_j that guarantees the satisfaction of that formula at s consists in all agents from A_j acting j at s, following their strategy that guarantees in M_i the satisfaction of the objective Φ_j if the play enters the copy of M_i, and acting in an arbitrarily fixed manner at all other states of M. (Note that, if the strategy for A_j in M_i is positional, then the above described strategy is positional, too.) Lastly, every \([[B_i]])\times Ψ_i ∈ Γ\) is true at s, too, because if B_i ≠ ∅ then for every collective action of all agents from \([[B_i]])\times T\) there is a suitable complementary action of ∅ \ B_i, where all agents choose actions greater than m and such that neg(σ) adds up to l modulo n. (In fact, this can be guaranteed by any agent in ∅ \ B_i after all others have chosen their actions.) In the case when B_i = ∅, every subset \{ ⟨A_j⟩ \times Φ_j, [[B_i]] \times Ψ_i \} for j < m is of distributed control, hence Ψ_i is true at the root s_i of M_i for each i = 1, ..., d. Thus, M, s |= Γ, which completes the proof.

A consequence of the proof above is that memoryless and memory-based semantics yield the same satisfiable state formulae in the flat fragments.

**Corollary 4.10** A St(ATL₁) formula Φ is satisfiable in the memoryless semantics if and only if it is satisfiable in the memory-based semantics.

**Proof.** Lemma 4.9 can be proved for memoryless semantics in the same way, but only for St(ATL₁) formulae. This is because the successor components are LTL formulae which have the same semantics with and without memory. Further, for both semantics each subformula \(⟨A⟩\theta\) or \([[A]]\theta\) of Φ with a strategic quantifier as main connective can be converted to SNF by converting θ to LSNF using Lemma 4.6. Then, we can use the memoryless and memory-based version of Lemma 4.9 and obtain that Φ is satisfiable in the memory-based semantics if and only if it is satisfiable in the memoryless semantics since the satisfiable sets of successor components are the same for the two types of semantics.

## 5 Optimal decision procedures for satisfiability in fragments of ATL₁

### 5.1 Centipede models. Satisfiability in LTL₁, CTL₁ and CTL⁺

Satisfiability of LTL₁ is analyzed in [5]. In particular, it is shown that if an LTL₁ formula θ is satisfiable then it is satisfiable in a model of the form s_0s_1...s_ℓ where ℓ = |θ|. Consequently, it is shown that satisfiability of LTL₁ is NP-complete. We provide similar results for CTL⁺ and CTL₁ here.
Proposition 5.1 If a CTL\(^+\) formula \(\varphi\) has a model, then it has a model with at most \(O(|\varphi|^2)\) states.

Proof. Suppose \(M, s_0 \models \varphi\) for a CTL\(^+\) formula \(\varphi\), a model \(M\) and a state \(s_0\). Assume w.l.o.g. that \(\varphi\) is in NNF. We generate another model \(M'\) with \(O(|\varphi|^2)\) states and a state \(s'_0\) such that \(M', s'_0 \models \varphi\). Let \(\Delta_Q\) be the set of subformulae of \(\varphi\) that has \(Q\) as main connective for \(Q \in \{E,A\}\) and let \(\Delta_B\) be the set of maximal Boolean subformulae of \(\varphi\) that do not occur in the scope of a path quantifier. For each \(Z \in \{E,A,B\}\) let \(\Delta^+_{\Delta} \subseteq \Delta_Z\) be the subsets satisfied in \(M, s_0\). Now, for each \(E\psi \in \Delta^+_E\) let \(\rho^\psi = \rho_0^\psi \rho_1^\psi \ldots\) be a path in \(M\) starting in \(s_0\) such that \(\rho^\psi \models \psi\). Since \(M, s_0 \models \varphi\) we have for every \(A\psi' \in \Delta^+_A\) that \(\rho^\psi \models \psi'\) because \(\psi'\) is satisfied along all paths from \(s_0\). Further, \(\rho_0^\psi = s_0\) implies that \(\rho^\psi \models \psi \land \bigwedge_{\psi' \in \Delta^+_A} \psi' \land \bigwedge_{\beta \in \Delta^+_B} \beta\). Since this is an LTL formula of size at most \(|\varphi|\) it has a model \(\pi^\psi\) of the form \(\pi_0^\psi \ldots (\pi_{|\varphi|})^\psi\) where \(\pi_0^\psi\) is labelled as \(s_0\). Now, by gluing together each path \(\pi^\psi\) (which is made finite by adding a self-loop to the state \(\pi_0^\psi\)) in the initial state \(s_0\) we obtain a transition system \(M'\) such that \(M', s'_0 \models \varphi\). Since \(|\Delta^+_E| \leq |\varphi|\) there are at most \(O(|\varphi|^2)\) states in \(M'\). \(\square\)

Further, we will see that for satisfiable formulae of CTL\(^+\), ATL\(_1\) and St(ATL\(^+\)) there are models that can be obtained by gluing together ultimately periodic paths as in the proof of Proposition 5.1. We call such models centipede models, illustrated in Figure 2. Note that these models only branch in the initial state.

![Fig. 2. A centipede model](image-url)

However, for the flat fragments CTL\(^+\), ATL\(_1\), St(ATL\(^+\)) models of polynomial size are not guaranteed to exist as for CTL\(^+\). First, the length of the period and the prefix of the ultimately periodic paths can be exponential due to LTL subformulae in the case of CTL\(^+\) and St(ATL\(^+\)). Second, in the cases of ATL\(_1\) and St(ATL\(^+\)) (but not for CTL\(^+\)) an exponential branching factor in the initial state may be forced by a formula. Indeed, consider the following ATL\(_1\) formula

\[
\varphi = \bigwedge_{i=1}^n [i]X p_i \land [i]X \neg p_i
\]

over the propositions \(\{p_1, \ldots, p_n\}\) and players \(\{1, \ldots, n\}\). For a state \(s_0\) to satisfy this formula there has to be a successor state for each possible truth assignment.
Proposition 5.2. Satisfiability in $\text{CTL}_1^+$ is NP-complete.

Proof. NP-hardness follows directly from Boolean satisfiability. An NP-algorithm for $\text{CTL}_1^+$ works as follows. It takes as input a $\text{CTL}_1^+$ formula $\varphi$ in NNF, hence a positive Boolean combination of flat $\text{CTL}_1^+$ state formulae, and guesses non-deterministically a centipede model $M$, $s_0$ of size $O(|\varphi|^2)$ for $\varphi$, as well as the disjuncts in $\varphi$ that evaluate to true at $s_0$. (According to Proposition 5.1, if $\varphi$ has a model then it has a model of this form and size.) After guessing, it checks whether the resulting formula of the form $\varphi' = \beta \land \bigwedge_{i=0}^k \varphi_i'$ is true in the guessed model where $\beta$ is a Boolean formula and each $\varphi_i'$ is of the form $A_\theta_i$ or $E_\theta_i$ for an $\text{LTL}_1$ formula $\theta_i$. First, the model-checking of $\beta$ can be done in linear time. Next, for each of the $O(|\varphi|)$ formulae $\theta_i$ it can checked whether it is true in each of the $O(|\varphi|)$ paths of the centipede model in polynomial time since $\text{LTL}$ model-checking of an ultimately periodic path of length $O(|\varphi|)$ can be done in polynomial time in $|\varphi|$ [11]. Thus, the guess can be verified in polynomial time due to the small model property of Proposition 5.1 and the centipede shape of the model.

\[\square\]

Corollary 5.3. Satisfiability in $\text{CTL}_1$ is NP-complete.

5.2 Lower bound for satisfiability in $\text{ATL}_1$

Proposition 5.4. $\text{ATL}_1\text{-SAT}$ is $\Sigma_3^P$-hard.

Proof. The proof is by reduction from the $\Sigma_3^P$-SAT problem, which is $\Sigma_3^P$-complete. This problem takes as input a quantified Boolean sentence

$$\gamma = \exists x_1, \ldots, x_m \forall x_{m+1}, \ldots, x_n \exists x_{k+1}, \ldots, x_n \gamma'$$

where $\gamma'$ is a Boolean formula over the Boolean variables $x_1, \ldots, x_n$. The output is true if and only if $\gamma$ is true. Given $\gamma$, we construct an $\text{ATL}_1$ formula $\psi(\gamma)$ over the set $\text{Prop} = \{x_1, \ldots, x_n\}$ of proposition symbols as follows

$$\psi = \bigwedge_{i=1}^m (AX x_j \lor AX \neg x_j) \land \bigwedge_{i=m+1}^n (\langle\{i\}\rangle x_i \lor \langle\{i\}\rangle \neg x_i) \land \neg \langle\{m+1, \ldots, k\}\rangle X \neg \gamma'$$

We now claim that $\gamma$ is true if and only if $\psi(\gamma)$ is satisfiable.

First, suppose that $\gamma$ is true. Then we construct a CGM $M = (A, \text{St}, \{\text{Act}_a\}_{a \in A}, \{\text{Act}_s\}_{a \in A}, \text{out}, \text{Prop}, \text{L})$ and a state $s_0 \in \text{St}$, such that $M, s_0 \models \psi(\gamma)$, as follows. Let $A = \{m + 1, \ldots, n\}$, $\text{St} = \{s_0\} \cup \{s_{m+1}, \ldots, s_n \mid v_i \in \{0, 1\} \text{ for } m + 1 \leq i \leq n\}$, $\text{Act}_a = \{0, 1\}$ for all $a \in A$. Then, for every agent $a \in A$ define $\text{act}_a(s_0) = \{0, 1\}$ and $\text{act}_a(s) = \{0\}$ for all $s \neq s_0$. The transitions are defined by $\text{out}(s_0, \langle v_{m+1}, \ldots, v_n \rangle) = s_{v_{m+1}, \ldots, v_n}$ and $\text{out}(s, \alpha_A) = s$ for all $s \neq s_0$ and all action profiles $\alpha_A$. The set of proposition symbols is $\text{Prop} = \{x_1, \ldots, x_n\}$. The labelling is given by $L(s_0) = \emptyset$ and for every
\(i \in \{m+1, \ldots, n\}\) we have \(x_i \in L(s_{v_{m+1}, \ldots, v_n})\) if and only if \(v_i = 1\) when \(s_{v_{m+1}, \ldots, v_n} \in \text{St} \setminus \{s_0\}\). Finally, let \(x'_1, \ldots, x'_m\) be particular values such that

\[
\forall x_{m+1}, \ldots, x_k \exists x_{k+1}, \ldots, x_n. \gamma'[x_1 \mapsto x'_1, \ldots, x_m \mapsto x'_m] \text{ is true.}
\]

Such values exist since \(\gamma\) is true. For \(1 \leq i \leq m\) and every \(s_{v_{m+1}, \ldots, v_n} \in \text{St} \setminus \{s_0\}\), let \(x_i \in L(s_{v_{m+1}, \ldots, v_n})\) if and only if \(x'_i = 1\).

Intuitively, for all \(i\) such that \(m+1 \leq i \leq n\) player \(i\) chooses the value of \(x_i\) in the successor state and then the play stays in that state forever. The value of \(x_i\) for \(1 \leq i \leq m\) in the successor state is defined by the values \(x'_1, \ldots, x'_m\). The subformula \(\bigwedge_{i=m+1}^n (\langle \{i\} \rangle X x_i \land \langle \{i\} \rangle \neg x_i)\) is clearly true at \(s_0\). The same is the case for \(\bigwedge_{i=1}^m (AX x_i \lor AX \neg x_i)\). Next, since \(\gamma\) is true when \(x_i\) takes the values \(x'_i\) for \(1 \leq i \leq m\), then no matter which values of \(x_i\) are chosen by players in \(\{m+1, \ldots, k\}\) there exists values of \(x_i\) for players in \(\{k+1, \ldots, n\}\) such that \(\gamma'\) is true in the successor state. Thus, coalition \(\{m+1, \ldots, k\}\) does not have a strategy to ensure that \(\gamma'\) is false in the successor state. Thus, \(\mathcal{M}, s_0 \models \psi(\gamma)\).

For the converse direction, suppose that \(\psi(\gamma)\) is satisfied by some model \(\mathcal{M}, s_0\). For contradiction, suppose that \(\gamma\) is false. Then for all \(x_1, \ldots, x_m\) there exists \(x_{m+1}, \ldots, x_k\) such that \(\gamma'\) is false for all \(x_{k+1}, \ldots, x_n\). In particular, this must be the case when \(x_i\) take the unique values \(x'_i\) for \(1 \leq i \leq m\) that are true in all successors of \(s_0\). These are unique since \(s_0\) satisfies \(\bigwedge_{j=1}^m (AX x_j \lor AX \neg x_j)\).

In this case there exists particular values \(x'_i\) for \(m+1 \leq i \leq k\) such that \(\gamma'\) is false for all \(x_{m+1}, \ldots, x_k\) when \(x_i\) take the values \(x'_i\) for \(m+1 \leq i \leq k\). Consider the strategy for coalition \(\{m+1, \ldots, k\}\) that chooses these values for \(x_i\) in the successor state for \(m+1 \leq i \leq k\). This strategy ensures that \(\gamma'\) is false in the successor state. However, this contradicts the fact that \(\mathcal{M}, s_0 \models \neg \langle \{m+1, \ldots, k\} \rangle X \neg \gamma'\). Thus, \(\gamma\) must be true. This completes the proof. \(\square\)

Note that the hardness result only requires the use of the temporal operator \(X\) and neither \(U\) nor \(R\). This is interesting since this lower bound will be shown to be an upper bound for the full \(\text{ATL}^+\) in the following section. Thus, the \(\langle \{i\} \rangle X\) fragment of \(\text{ATL}^+_1\) is as hard as the full \(\text{ATL}^+_1\).

### 5.3 Deciding satisfiability in \(\text{St(\text{ATL}^+_1)}\) and \(\text{ATL}^+\)

**Lemma 5.5** Let \(\Phi = \langle \langle C \rangle \rangle \Psi\) be an \(\text{ATL}^*\) formula and let \(\text{Prop}(\Phi) = \{p_1, \ldots, p_r\}\) be the set of atomic propositions occurring in \(\Phi\). Consider any mapping \(v : \text{Prop}(\Phi) \to \{\top, \bot\}\) and let \(v[\Phi]\) be the result of substitution of all occurrences of \(p_i\) in \(\Phi\) which are not in the scope of a temporal operator by \(v(p_i)\), for each \(p_1, \ldots, p_r\). Further, let

\[
\delta(v) := \bigwedge_{v(p_i) = \top} p_i \land \bigwedge_{v(p_i) = \bot} \neg p_i
\]

Then, \(\delta(v) \land \Phi \equiv \delta(v) \land v[\Phi]\).

**Proof.** Consider any CGM \(\mathcal{M}\) and a state \(s\) in it. If \(\delta(v)\) is false at \(s\) then both sides are false. Suppose \(\mathcal{M}, s \models \delta(v)\). Then \(\mathcal{M}, s \models v(p_i) \leftrightarrow p_i\) for each \(p_1, \ldots, p_r\). Then, \(\Phi\) and \(v[\Phi]\) are equally true or false at \(s\), as they only differ in the occurrences of atomic propositions that are evaluated at \(s\). \(\square\)
Proposition 5.6  
(i) The satisfiability testing for \( \text{St}(\text{ATL}_1^+) \) is in \( \text{PSPACE} \).

(ii) The satisfiability testing for \( \text{ATL}_1^+ \) (and \( \text{ATL}_1 \)) is in \( \Sigma_3^P \).

**Proof.** The decision procedures for both \( \text{St}(\text{ATL}_1^+) \) and \( \text{ATL}_1^+ \) will essentially the same, but in their last phases they work in different computational complexities. First, consider an \( \text{St}(\text{ATL}_1^+) \) formula \( \Phi \) and let \( \text{Prop}(\Phi) = \{ p_1, \ldots, p_r \} \). The formula \( \Phi \) is a Boolean combination of atomic propositions and subformulae of the type \( \langle C \rangle \theta \) where \( \theta \in \text{LTL} \). By Lemma 4.6, we can assume that each such \( \theta \) is in a LSNF of linearly increased length, i.e., is a Boolean combination of atomic propositions and \( \text{X} \)-formulae (formulae beginning with \( \text{X} \)) of \( \text{LTL} \).

The algorithm now works as follows:

1. Guess a truth assignment \( \tau \) for the atomic propositions in \( \text{Prop}(\Phi) \) at a state \( s \) of a CGM satisfying \( \Phi \), if any. Consider the unique map \( v : \text{Prop}(\Phi) \rightarrow \{ \top, \bot \} \) for which \( \delta(v) \) is true under \( \tau \). By Lemma 5.5, each maximal subformula \( \langle C \rangle \theta \) in \( \Phi \) is equivalently replaced by \( v[\langle C \rangle \theta] \), which is \( \langle C \rangle v[\theta] \).

2. After elementary Boolean simplifications (of the type \( \top \land A \equiv A, \bot \land A \equiv \bot, \text{etc.} \)) each \( v[\theta] \) is transformed to a Boolean combination of \( \text{X} \)-formulae only. Using the \( \text{LTL} \) validities \( X \eta \land X \xi \equiv X(\eta \land \xi) \) and \( X \eta \lor X \xi \equiv X(\eta \lor \xi) \), it is further equivalently transformed into an \( \text{X} \)-formula which is at most as long.

The original formula is now (non-deterministically) transformed to an equisatisfiable Boolean combination of \( \text{ATL}_1^+ \) formulae of type \( \langle C \rangle X \theta \) and \( [C]X \theta \).

3. Now, assuming that the resulting formula is satisfiable, we further guess the true disjuncts in every \( \mathbf{\lor} \)-subformula in a satisfying CGM and reduce the problem to checking satisfiability of a conjunctive formula of the type

\[
\Theta = \langle A_0 \rangle X \theta_0 \land \ldots \land \langle A_{m-1} \rangle X \theta_{m-1} \land [[B_m]] X \eta_0 \land \ldots \land [[B_{n-1}]] X \eta_{n-1} \n\]

Let \( D(\Theta) \) be the union of the set \( C(\Theta) \) of conjuncts of \( \Theta \) and \( \{ [[A]] X \top \} \), i.e.

\[
D(\Theta) = \{ \langle A_0 \rangle X \theta_0, \ldots, \langle A_{m-1} \rangle X \theta_{m-1}, [[B_m]] X \eta_0, \ldots, [[B_{n-1}]] X \eta_{n-1}, [[A]] X \top \}
\]

4. By Lemma 4.9, the set \( D(\Theta) \) is satisfiable iff every subset of distributed control of it has a satisfiable set of successor components. Since each of them is a set of \( \text{LTL} \) formulae, these checks can be done using standard techniques.

Each check in step 4. of the algorithm can be done in \( \text{PSPACE} \) when \( \Phi \) is a \( \text{St}(\text{ATL}_1^+) \) formula, since each successor component is an \( \text{LTL} \) formula. In the case of \( \text{ATL}_1^+ \) the checks can be done in \( \text{NP} \) according to [5], as in this case each successor component is an \( \text{LTL}_1 \) formula. Hence, checking that each of the (possibly exponentially many) subsets of distributed control is satisfiable can be done in \( \text{coNP}^{\text{PSPACE}} = \text{PSPACE} \) for \( \text{St}(\text{ATL}_1^+) \) and in \( \text{coNP}^{\text{NP}} \) for \( \text{ATL}_1^+ \). Thus, the whole procedure can be done respectively in \( \text{NP}^{\text{PSPACE}} = \text{PSPACE} \) for \( \text{St}(\text{ATL}_1^+) \) and in \( \text{NP}^{\text{coNP}^{\text{NP}}} \) for \( \text{ATL}_1^+ \), by guessing the true propositions in the initial state and the true disjuncts in \( \Phi \), and then applying resp. a \( \text{PSPACE} \)-oracle and \( \text{coNP}^{\text{NP}} \)-oracle. Since \( \text{NP}^{\text{coNP}^{\text{NP}}} = \Sigma_3^P \) the proof is completed. \( \square \)
This result, combined with Proposition 5.4 and the PSPACE-hardness of LTL satisfiability, yields the following.

**Theorem 5.7** The satisfiability problem of
(i) \(\text{St}(\text{ATL}^*_1)\) is PSPACE-complete
(ii) \(\text{CTL}^*_1\) is PSPACE-complete
(iii) \(\text{ATL}^+_1\) is \(\Sigma^P_3\)-complete
(iv) \(\text{ATL}^*_1\) is \(\Sigma^P_3\)-complete

Here is another consequence of the proof of Proposition 5.6:

**Corollary 5.8** Every satisfiable \(\text{St}(\text{ATL}^*_1)\) formula \(\Phi\) has a centipede model \(M\) with branching factor \(O(2^{\|\Phi\|})\) in the root. Further, every ultimately periodic path in \(M\) has a prefix of length \(O(2^{\|\Phi\|})\) and a period of length \(O(|\Phi| \cdot 2^{\|\Phi\|})\).

### 5.4 PSPACE decision procedure for the satisfiability in \(\text{ATL}^*_1\)

The decision procedure for \(\text{St}(\text{ATL}^*_1)\) can be extended to a PSPACE-complete decision procedure for the whole \(\text{ATL}^*_1\), by combining it with a PSPACE decision procedure for LTL and showing that every path-satisfiable \(\text{ATL}^*_1\) formula can be satisfied in a special type of CGMs described below. The proof of the latter is rather lengthy (see brief discussion further), so we only state and prove here the easier case of the slightly smaller fragment \(\hat{\text{ATL}}^*_1\), where no strategic quantifiers \(\langle\langle \emptyset \rangle\rangle\) (i.e., fully universal path quantifiers) are allowed. We only note that the procedure for the full \(\text{ATL}^*_1\) is essentially the same.

First, recall that every satisfiable LTL formula has an ultimately periodic linear model with prefix and period that both have length exponential in the size of the formula [14]. Further, according to Corollary 5.8, every satisfiable \(\text{St}(\text{ATL}^*_1)\) formula can be satisfied at the root state of a centipede model of exponentially bounded number and length of legs. Combining these results leads to a new type of CGMs which we call *Lasso of Centipedes (LoC) models*. Such models consist of an ultimately periodic path (the lasso) where each state is the root of a centipede model. An illustration of a model like this is shown in Figure 3.

**Proposition 5.9** Every satisfiable \(\hat{\text{ATL}}^*_1\) formula \(\Phi\) is satisfied in a LoC model with size bounded exponentially in \(|\Phi|\).

**Proof.** Given an \(\hat{\text{ATL}}^*_1\) formula \(\Phi\) we define its LTL skeleton \(\text{Sk}_{\text{LTL}}(\Phi)\) as follows: Let the state subformulae of \(\Phi\) of type \(\langle\langle C \rangle\rangle\theta\) or \(\llbracket C \rrbracket\theta\) be \(\Psi_1, \ldots, \Psi_n\). For each of them \(\Psi\) we introduce a new (not in \(\text{Prop}\)) atomic proposition \(p_\Psi\). Then we produce the LTL formula \(\hat{\Phi}\) by replacing every occurrence of such a subformula \(\Psi\) in \(\Phi\) by \(p_\Psi\). Now, define

\[
\text{Sk}_{\text{LTL}}(\Phi) := \hat{\Phi} \land \bigwedge^n_{i=1} G (p_{\Psi_i} \rightarrow \Psi_i)
\]
We claim that any CGM $M$ and a path $\pi$ in it on which $\Phi$ is true can be expanded to a CGM $\hat{M}$ and a path $\hat{\pi}$ in it satisfying $\text{Sk}_{\text{LTL}}(\Phi)$, by evaluating each new atomic proposition $p_\Psi$ to be true at exactly those states of $\pi$ at which $\Psi$ is true in $M$. Conversely, for any CGM $M$ and a path $\pi$ in it on which $\text{Sk}_{\text{LTL}}(\Phi)$ is true, the formula $\Phi$ is true on $\pi$, too, because all atomic propositions $p_\Psi$, occur only positively in $\hat{\Phi}$, so replacing them with the respective $\Psi_i$'s will preserve the truth.

Thus, it suffices to show that if $\text{Sk}_{\text{LTL}}(\Phi)$ is path-satisfiable then it can be satisfied on the lasso path in some LoC model of size bounded exponentially in $|\Phi|$. Indeed, take any CGM $M$ and a path $\pi$ in it on which $\text{Sk}_{\text{LTL}}(\Phi)$ is true. Then, in particular, the path $\pi$ alone is a linear model for $\hat{\Phi}$. Now, take an ultimately periodic linear model $\hat{\pi}$ of length bounded exponentially in $|\hat{\Phi}|$, hence in $|\Phi|$. Such a model can be obtained from $\pi$ by cutting its tail off at a suitable position and looping back to a suitable previous state. Thus, every state in $\hat{\pi}$ has the label of a prototype state in $\pi$. Now, for every state $\hat{s}$ on $\hat{\pi}$, let $s$ be its prototype in $\pi$. We do the following.

- Consider the set $\Gamma(s)$ of state subformulae $\Psi$ of $\Phi$ such that $p_\Psi$ is in the label of $s$ in $\pi$. Since $\text{Sk}_{\text{LTL}}(\Phi)$ is true on $\pi$, every formula in $\Gamma(s)$ is true at $s$ in $M$. Thus, $\Gamma(s)$ is satisfiable, hence by Corollary 5.8, it can be satisfied at the root state of a centipede model $M(\Gamma(s))$ of exponentially bounded in $|\Phi|$ number and length of legs.

- Now, we graft a copy of $M(\Gamma(s))$ at the state $\hat{s}$ in $\hat{\pi}$ by identifying its root with $\hat{s}$ and keeping all other states disjoint from $\hat{\pi}$.

- Next, we add a special new action for every agent at the state $\hat{s}$ and define the successor of the resulting action profile to be the successor of $\hat{s}$ on the path $\hat{\pi}$, while every other action profile involving some (but not all) of these special new actions leads to a successor of $\hat{s}$ in the grafted copy of $M(\Gamma(s))$, chosen so as not to affect the truth of any of the formulae from $\Gamma(s)$ at $\hat{s}$.

We omit the easy but tedious details of this construction.

After completing this procedure for each state of $\hat{\pi}$, the result is a LoC model $\hat{M}$ which, by construction, satisfies the formula $\hat{\Phi}$ on $\hat{\pi}$ and satisfies at each state $\hat{s}$ on $\hat{\pi}$ the set $\Gamma(s)$. Therefore, $\hat{M}, \hat{\pi} \models \text{Sk}_{\text{LTL}}(\Phi)$, hence $\hat{M}, \hat{\pi} \models \Phi$, $\square$
For lack of space we only briefly indicate the additional complication in extending this result to $\text{ATL}_1^\star$: if a subformula $\Psi = \langle\langle\emptyset\rangle\rangle\theta$ is true at some state of the path $\pi$ in the CGM satisfying $\Phi$, its effect cannot be constrained only on the centipede model grafted at the respective state of $M$, as done above, but it propagates through the path $\tilde{\pi}$ to all centipede models grafted at all further states on $\tilde{\pi}$. So, additional description in LTL is needed to describe and preserve this effect when converting $\pi$ into the lasso $\tilde{\pi}$. That is why we state the next result relativized to only what we have proved here.

Proposition 5.10 The path-satisfiability problem in LoC models of size bounded exponentially in the length of input $\text{ATL}_1^\star$ formulae is in PSPACE.

Proof. The algorithm begins like the PSPACE decision procedure for LTL satisfiability that guesses the lasso on the fly for an LTL input formula $\theta$ [14]. First, the length of the prefix and the length of the period are guessed. At each step around the lasso, the subformulae that are true from the current state are guessed non-deterministically and a local consistency check as well as a one-step consistency check are performed. Further, a set $\Delta$ (of at most polynomial size) of eventuality formulae is kept to make sure that all eventualities that are needed for $\theta$ to be true are actually true further on the lasso.

The algorithm for $\text{ATL}_1^\star$ works in the same way on an $\text{ATL}_1^\star$ formula $\Phi$, but treats strategically quantified subformulae of $\Phi$ as atomic propositions and, at each step of the procedure, the local consistency check includes verifying these subformulae that have to be true at the current state. This amounts to checking satisfiability of an $\text{St}(\text{ATL}_1^\star)$ formula and can be done in PSPACE, by Theorem 5.7. For the formulae of the form $\langle\langle A \rangle\rangle\theta$ where $A \neq \emptyset$ this can be done independently of the rest of the lasso, by ensuring that when agents in $A$ commit to satisfying $\theta$ then the play goes into the centipede (and stays there). But, when $A = \emptyset$ then $\theta$ has to be true on all paths from the current state. This includes both the path around the lasso and those that enter one of the centipedes at some point. Note that the original set $\Delta$ of formulae we are keeping only needs to be satisfied around the lasso. To keep track of this we keep, in addition to $\Delta$, an extra set of formulae $\Gamma$ which must be satisfied both around the lasso and on paths that exit to a centipede. Thus, the formulae in $\Gamma$ must be included in the $\text{St}(\text{ATL}_1^\star)$ satisfiability check at each step. But since $\Gamma$ is polynomial in size at each step, this check can still be performed in PSPACE. This means that the entire procedure can be performed in PSPACE.

Corollary 5.11 Satisfiability of $\hat{\text{ATL}}_1^\star$ is PSPACE-complete.

6 Concluding remarks and summary of results

We have developed optimal decision procedures for the satisfiability problems in flat fragments of $\text{ATL}^\star$, and in particular $\text{CTL}^\star$ and have obtained exact complexity results for them. A summary of the main complexity results obtained in this paper is provided in the table in Fig. 4. It shows that these complexities are much lower than those for the full languages while, in view of Proposition 3.2, they are very tight with respect to syntactic extensions in terms of nesting
depth of formulae.

<table>
<thead>
<tr>
<th>$L$</th>
<th>SAT($L$)</th>
<th>SAT($L_1$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTL$^+$</td>
<td>EXPTIME [6]</td>
<td>NP (Cor. 5.3)</td>
</tr>
<tr>
<td>CTL$^*$</td>
<td>2EXPTIME [10]</td>
<td>NP (Prop. 5.2)</td>
</tr>
<tr>
<td>ATL</td>
<td>EXPTIME [16]</td>
<td>$\Sigma_3^P$ (Theo. 5.7)</td>
</tr>
<tr>
<td>ATL$^+$</td>
<td>2EXPTIME [12][10]</td>
<td>$\Sigma_3^P$ (Theo. 5.7)</td>
</tr>
<tr>
<td>ATL$^*$</td>
<td>2EXPTIME [12]</td>
<td>PSPACE (Theo. 5.7, Cor. 5.11)</td>
</tr>
</tbody>
</table>

Fig. 4. Complexity of satisfiability. All results are completeness results. In the case of ATL$^*$ the results refer to $St(\text{ATL}_1^1)$ and $\hat{\text{ATL}}_1^1$.

References


