

Optimal Decision Procedures for Satisfiability in Fragments of Alternating-time Temporal Logics

Valentin Goranko^{a,b} Steen Vester^a¹

^a*Department of Applied Mathematics and Computer Science
Technical University of Denmark*

^b*Department of Mathematics, University of Johannesburg*

Abstract

We consider several natural fragments of the alternating-time temporal logics ATL* 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* is PSPACE-complete, whereas the satisfiability problems in the flat fragments of ATL and ATL⁺ are Σ_3^P -complete. We note that the nesting hierarchies for fragments of ATL* 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* was introduced and studied in [2] as a multi-agent extension of the branching time temporal logic CTL*, applied for specification and verification of properties of open systems. The most natural semantics for ATL* 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* 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 Φ on all plays (computations) enabled by that collective strategy. The logic ATL* 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*, and even its fragment ATL, turned out to have some problematic semantic features

¹ Emails: vfgo@dtu.dk, stve@dtu.dk

related to the nesting of strategic quantifiers $\langle\langle\cdot\rangle\rangle$, including:

- Conceptual difficulty in understanding the very meaning of nested expressions of the type $\langle\langle A \rangle\rangle \dots \langle\langle B \rangle\rangle \Phi$, especially, when the coalitions A and B share common agents. For instance, what exactly should $\langle\langle A \rangle\rangle \neg \langle\langle A \rangle\rangle \Phi$ mean?
- This problem is related to a technical problem built in the semantics of ATL^* , where e.g., in the truth evaluation of a formula of the type $\langle\langle A \rangle\rangle \dots \langle\langle B \rangle\rangle \Phi$ the strategy for A adopted to guarantee the success of the goal $\dots \langle\langle B \rangle\rangle \Phi$ does not have any effect when evaluating the truth of the subgoal $\langle\langle B \rangle\rangle \Phi$, which, arguably, goes against the intuitive understanding of what a strategy and its execution mean. Such problems have led to several proposals of alternative semantics for 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 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 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 ATL becomes undecidable [2].

So, there are several good reasons to consider *flat* fragments of 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 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 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 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 ATL^* .
- (ii) The algorithmic problem of satisfiability testing in the flat fragments of ATL and ATL^+ , where only nesting of temporal operators in the scope of strategic quantifiers is allowed, are Σ_3^P -complete, in contrast to the

2EXPTIME-completeness of that problem in the full ATL^+ (as subsuming CTL^+ , see [10]) and its EXPTIME-completeness in the full ATL [8].

The structure of the paper is as follows: In Section 2 we summarize basics of the logics LTL , CTL and CTL^* as well as concurrent game models and the alternating-time temporal logics ATL , ATL^+ and ATL^* . In Section 3 we introduce various flat fragments of ATL^* and discuss their expressiveness. Section 4 contains the technical preparation for our algorithms, where we introduce some kinds of normal forms for 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 ATL^* considered in the paper. We end with brief concluding remarks in Section 6.

2 Preliminaries

2.1 Summary of LTL , CTL , CTL^* and their flat fragments

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

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

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

$$\theta U \eta \equiv \eta \vee (\theta \wedge X(\theta U \eta)), \quad \theta R \eta \equiv \eta \wedge (\theta \vee X(\theta R \eta)).$$

We define the flat fragments LTL_1 , CTL_1 and CTL_1^* resp. as subsets of LTL , CTL and CTL^* . In LTL_1 no nesting of temporal operators is allowed, in CTL_1^* no nesting of path quantifiers is allowed and in CTL_1 neither is allowed. They are generating as follows, where β is a Boolean formula and θ is an LTL formula:

$$\begin{aligned} \text{LTL}_1: \quad \theta &::= p \mid \neg\theta \mid \theta \wedge \theta \mid X\beta \mid \beta U \beta; \\ \text{CTL}_1^*: \quad \varphi &::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid A\theta. \\ \text{CTL}_1: \quad \varphi &::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid AX\beta \mid A(\beta U \beta) \mid A(\beta R \beta). \end{aligned}$$

For instance:

- $p U q \wedge X(r \wedge (q \wedge \neg p))$ is in LTL_1 but $p U (Xq)$ is not.
- $A(\neg p U (p \wedge \neg q)) \wedge \neg(EF(q \wedge \neg p) \wedge \neg AF \neg(p \wedge q))$ is in CTL_1 (and in CTL_1^*).
- $AG F p$ is in CTL_1^* but not in CTL_1 ; $AGE F p$ is neither in CTL_1 nor in CTL_1^* .

2.2 Concurrent game models. The logic ATL* and fragments

A **concurrent game model** [2] (CGM) is a tuple

$\mathcal{M} = (\mathbb{A}, \text{St}, \{\text{Act}_a\}_{a \in \mathbb{A}}, \{\text{act}_a\}_{a \in \mathbb{A}}, \text{out}, \text{Prop}, \text{L})$ comprising:

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

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 $s_0 s_1 \dots \in \text{St}^\omega$ of states such that for each $i \geq 0$ there exists an action profile $\alpha_{\mathbb{A}} = \langle \alpha_1, \dots, \alpha_k \rangle$ such that $\text{out}(s_i, \alpha_{\mathbb{A}}) = s_{i+1}$. A *history* is a finite initial segment $s_0 s_1 \dots s_\ell$ of a play. We denote by $\text{Play}_{\mathcal{M}}$ and $\text{Hist}_{\mathcal{M}}$ respectively the set of plays and set of histories in \mathcal{M} . For a state $s \in \text{St}$ we define $\text{Play}_{\mathcal{M}}(s)$ and $\text{Hist}_{\mathcal{M}}(s)$ as the set of plays and set of histories with initial state s . For a sequence ρ of states ρ_0 is the initial state, ρ_i is the $(i+1)$ th state, $\rho_{\leq i}$ is the prefix $\rho_0 \dots \rho_i$ of ρ and $\rho_{\geq i}$ is the suffix $\rho_i \rho_{i+1} \dots$ of ρ . When $\rho = \rho_0 \dots \rho_\ell$ is finite, we say that it has length ℓ and write $|\rho| = \ell$. Further, we let $\text{last}(\rho) = \rho_\ell$.

A *strategy* for a player a in \mathcal{M} is a mapping $\sigma_a : \text{Hist}_{\mathcal{M}} \rightarrow \text{Act}_a$ such that for all $h \in \text{Hist}_{\mathcal{M}}$ we have $\sigma_a(h) \in \text{act}_a(\text{last}(h))$. 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 $\text{Strat}_{\mathcal{M}}(a)$ the set of strategies of player a . A (collective) strategy of a coalition $C \subseteq \mathbb{A}$ is a tuple $(\alpha_a)_{a \in C}$ of strategies, one for each player in C . When $C = \mathbb{A}$ this is called a *strategy profile*. We denote by $\text{Strat}_{\mathcal{M}}(C)$ the set of collective strategies of coalition C . A play $\rho \in \text{Play}_{\mathcal{M}}$ is consistent with a strategy $\sigma_C \in \text{Strat}_{\mathcal{M}}(C)$ if for every $i \geq 0$ there exists an action profile $\alpha_{\mathbb{A}} = \langle \alpha_1, \dots, \alpha_k \rangle$ such that $\text{out}(\rho_i, \alpha_{\mathbb{A}}) = \rho_{i+1}$ and $\alpha_a = \sigma_a(\rho_{\leq i})$ for all $a \in C$. The set of plays with initial state s that are consistent with σ_C is denoted $\text{Play}_{\mathcal{M}}(s, \sigma_C)$. In particular, we define $\text{Play}_{\mathcal{M}}(s, \sigma_a) = \text{Play}_{\mathcal{M}}(s, \sigma_{\{a\}})$ 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 ATL^* is a formula of type $\langle\langle C \rangle\rangle\Phi$, 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, ATL^* is a multi-agent extension of the branching time logic CTL^* with *strategic quantifiers* $\langle\langle C \rangle\rangle$ indexed with sets (*coalitions*) C of players. There are two types of formulae in ATL^* , *state formulae*, that are evaluated at states, and *path formulae*, that are evaluated on plays. These are defined by mutual recursion as follows, where $C \subseteq \mathbb{A}$, $p \in \text{Prop}$:

$$\begin{aligned} \text{State formulae of } \text{ATL}^*: \quad \varphi &::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle C \rangle\rangle\Phi, \\ \text{Path formulae of } \text{ATL}^*: \quad \Phi &::= \varphi \mid \neg\Phi \mid \Phi \wedge \Phi \mid \text{X}\Phi \mid \Phi \text{U}\Phi \mid \Phi \text{R}\Phi. \end{aligned}$$

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 ATL^* only involving strategic quantifiers for the empty coalition $\langle\langle \emptyset \rangle\rangle$, identified with universal path quantifier A , and for the “grand coalition” of all players $\langle\langle \mathbb{A} \rangle\rangle$, identified with existential path quantifier E . Equivalently, by identifying all agents, CTL^* can be regarded as the 1-agent fragment of ATL^* . To keep the notation lighter, we will list the members of C in $\langle\langle C \rangle\rangle$ without using $\{\}$.

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

$$\text{Path formulae of } \text{ATL}^+: \quad \Phi ::= \varphi \mid \neg\Phi \mid \Phi \wedge \Phi \mid \text{X}\varphi \mid \varphi \text{U}\varphi \mid \varphi \text{R}\varphi$$

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

$$\text{Formulae of } \text{ATL}: \quad \varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle C \rangle\rangle\text{X}\varphi \mid \langle\langle C \rangle\rangle(\varphi \text{U}\varphi) \mid \langle\langle C \rangle\rangle(\varphi \text{R}\varphi)$$

The combined operators $\langle\langle C \rangle\rangle F\varphi$ and $\langle\langle C \rangle\rangle G\varphi$ are defined respectively as $\langle\langle C \rangle\rangle \top \text{U}\varphi$ and $\langle\langle C \rangle\rangle \perp \text{R}\varphi$.

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

$$\begin{aligned} \mathcal{M}, s &\models p && \text{if } p \in L(s) \\ \mathcal{M}, s &\models \neg\varphi_1 && \text{if } \mathcal{M}, s \not\models \varphi_1 \\ \mathcal{M}, s &\models \varphi_1 \wedge \varphi_2 && \text{if } \mathcal{M}, s \models \varphi_1 \text{ and } \mathcal{M}, s \models \varphi_2 \\ \mathcal{M}, s &\models \langle\langle C \rangle\rangle\Phi && \text{if there exist a collective strategy } \sigma_C \in \text{Strat}_{\mathcal{M}}(C), \\ &&& \text{such that } \mathcal{M}, \rho \models \Phi \text{ for all } \rho \in \text{Play}_{\mathcal{M}}(s, \sigma_C) \end{aligned}$$

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

$\mathcal{M}, \rho \models \varphi$	if $\mathcal{M}, \rho_0 \models \varphi$
$\mathcal{M}, \rho \models \neg\Phi_1$	if $\mathcal{M}, \rho \not\models \Phi_1$
$\mathcal{M}, \rho \models \Phi_1 \wedge \Phi_2$	if $\mathcal{M}, \rho \models \Phi_1$ and $\mathcal{M}, \rho \models \Phi_2$
$\mathcal{M}, \rho \models \mathbf{X}\Phi_1$	if $\mathcal{M}, \rho_{\geq 1} \models \Phi_1$
$\mathcal{M}, \rho \models \Phi_1 \mathbf{U}\Phi_2$	if $\exists k. \mathcal{M}, \rho_{\geq k} \models \Phi_2$ and $\forall j < k. \mathcal{M}, \rho_{\geq j} \models \Phi_1$
$\mathcal{M}, \rho \models \Phi_1 \mathbf{R}\Phi_2$	if $\forall k. \mathcal{M}, \rho_{\geq k} \models \Phi_2$ or $\exists k. \mathcal{M}, \rho_{\geq k} \models \Phi_1$ and $\forall j \leq k. \mathcal{M}, \rho_{\geq j} \models \Phi_1$

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

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 $\langle\langle a \rangle\rangle \mathbf{X} \top$, 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 β, γ ; LTL formulae by θ, η, ζ ; ATL formulae by φ, ψ ; and ATL^* formulae – both state and path – by Θ, Φ, Ψ ; 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, $C \subseteq \mathbb{A}$, β is any Boolean formula and θ is any LTL formula:

- (i) **Separated ATL***, denoted $\text{ATL}_{\text{Sep}}^*$, consists of those formulae of 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:

$$\Phi ::= \theta \mid \neg\Phi \mid (\Phi \wedge \Phi) \mid \langle\langle C \rangle\rangle\Phi$$

- (ii) **Full (strategically) flat ATL***, denoted ATL_1^* , consists of those formulae of 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:

$$\Phi ::= p \mid \neg\Phi \mid (\Phi \wedge \Phi) \mid \langle\langle C \rangle\rangle\theta \mid X\Phi \mid \Phi U \Phi \mid \Phi R \Phi$$

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

- (iii) **State fragment of ATL***, denoted $\text{St}(\text{ATL}_1^*)$, consists of the state formulae of ATL_1^* , i.e. those formulae of 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:

$$\Phi ::= p \mid \neg\Phi \mid (\Phi \wedge \Phi) \mid \langle\langle C \rangle\rangle\theta.$$

- (iv) **Flat ATL⁺** (or, **double-flat ATL***), denoted ATL_1^+ , consists of those formulae of 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 ATL_1^+ are generated as follows, where $\theta \in \text{LTL}_1$:

$$\Phi ::= p \mid \neg\Phi \mid (\Phi \wedge \Phi) \mid \langle\langle C \rangle\rangle\theta$$

- (v) **Flat ATL**, denoted ATL_1 , consists of those formulae of ATL_1^+ which are in ATL , i.e., in which strategic quantifiers are followed immediately by temporal operators. The formulae of ATL_1 are generated as follows:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \langle\langle C \rangle\rangle X\beta \mid \langle\langle C \rangle\rangle(\beta U \beta) \mid \langle\langle C \rangle\rangle(\beta R \beta)$$

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:

- $\langle\langle 1 \rangle\rangle G \neg p \wedge \neg \langle\langle 2 \rangle\rangle X(p \vee \neg q) \vee \langle\langle 1, 2 \rangle\rangle(p U \neg q)$ is in ATL_1 ;
- $\langle\langle 1 \rangle\rangle(G \neg p \wedge F q), \neg \langle\langle 1, 2 \rangle\rangle((p R \neg q) \vee (\neg p U q))$ are in ATL_1^+ but not in ATL_1 ;
- $\langle\langle 1 \rangle\rangle(G F \neg p \vee X \neg(p U \neg q))$ is in $\text{St}(\text{ATL}_1^*)$ but not ATL_1^+ ;
- $G \langle\langle 1, 2 \rangle\rangle F(\neg p \vee q)$ is in $\widehat{\text{ATL}}_1^*$ but not in $\text{St}(\text{ATL}_1^*)$;

the size of Φ . Since $\text{AG}(p_\Psi \leftrightarrow \Psi)$ is a CTL formula, this works for each logic $\mathcal{L} \in \{\text{CTL}, \text{CTL}^+, \text{CTL}^*, \text{ATL}, \text{ATL}^+, \text{ATL}^*\}$, while for LTL we use $\text{G}(p_\Psi \leftrightarrow \Psi)$. \square

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 ATL*-fragments

The lower complexity of the satisfiability in the flat fragments of 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:

- $\langle\langle \text{ctrl} \rangle\rangle \text{G } \neg \text{break}$ in ATL_1 specifies that a controller can make sure the system does not break no matter how the environment behaves,
- $\bigwedge_{i=1}^n \langle\langle \text{proc}_i \rangle\rangle \text{G F db_access}_i$ 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 θ is satisfied on all fair paths (where fairness is defined by LTL formula θ_{fair}).

The semantics of 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 ATL is unaffected by such restrictions, but differences occur in ATL* and even in 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 ATL_1^* .

4 Normal forms and satisfiability of special sets in ATL*

4.1 Negation normal form of ATL* formulae

Definition 4.1 An ATL* formula Φ is in a *negation normal form* (NNF) if negations in Φ may only occur immediately in front of atomic propositions.

We now define the dual $[[\cdot]]$ to the strategic quantifier $\langle\langle \cdot \rangle\rangle$ as usual: $[[C]] ::= \neg \langle\langle C \rangle\rangle \neg$. If we consider $[[\cdot]]$ as a primitive operator in ATL*, then every 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 $[[\cdot]]$ formally breaks the syntax of the fragments ATL and 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 $[[\cdot]]$, using the following equivalences: $[[C]]X\varphi \equiv \neg \langle\langle C \rangle\rangle X \neg \varphi$, $[[C]](\varphi U \psi) \equiv \neg \langle\langle C \rangle\rangle ((\neg \varphi) R (\neg \psi))$, $[[C]](\varphi R \psi) \equiv \neg \langle\langle C \rangle\rangle ((\neg \varphi) U (\neg \psi))$.

Hereafter we assume that the language ATL* and each of its fragments introduced above are formally extended with the operator $[[\cdot]]$ 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 $[[\cdot]]$ can be transformed to an equivalent formula in NNF. Furthermore, each of the fragments ATL , ATL_1 , ATL_1^+ , $\text{St}(\text{ATL}_1^*)$ and ATL_1^* , extended with $[[\cdot]]$, 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 $\langle\langle C \rangle\rangle X \Phi$ or $[[C]] X \Phi$.

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

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

we associate the set of its

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

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 ζ 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 ζ is already transformed to NNF (of length less than twice the original length). Consider all maximal subformulae of ζ of the types $(\theta U \eta)$ and $(\theta R \eta)$. Replace each of them with its LTL-equivalent fixpoint unfolding, respectively $\eta \vee (\theta \wedge X(\theta U \eta))$ and $\eta \wedge (\theta \vee X(\theta R \eta))$. Then, the same procedure is applied recursively to all respective subformulae θ , η 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. \square

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

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

With every such formula Θ we associate the set of its *successor conjuncts*:

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

4.3 Sets of distributed control of ATL^* formulae

Definition 4.8 [Set of distributed control] A set of ATL^* formulae Δ is a *set of distributed control* if $\Delta = \{ \langle\langle A_0 \rangle\rangle \Phi_0, \dots, \langle\langle A_{m-1} \rangle\rangle \Phi_{l-1}, [[B]] \Psi \}$ where the

coalitions A_0, \dots, A_{l-1} are pairwise disjoint, and $A_0 \cup \dots \cup A_{l-1} \subseteq B$.

Lemma 4.9 *A set of ATL* successor formulae*

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

is satisfiable if and only if every subset of distributed control Δ of Γ has a satisfiable set of successor components.

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

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

Conversely, suppose that $\Delta_1, \dots, \Delta_d$ are all subsets of Γ of distributed control and they are all satisfiable. For each Δ_i we fix a CGM \mathcal{M}_i and a state s_i in it that satisfies $SC(\Delta_i)$. We can assume, w.l.o.g., that \mathcal{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 Γ 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 Γ . We will show that it suffices to take care of the sets of successor components of each subset of distributed control Γ and then will use the CGMs satisfying these to complete the construction of the model satisfying Γ .

Now, the construction. Recall that $|\mathbb{A}| = k$ and let $r = m + n$ (the numbers of $\langle\langle \cdot \rangle\rangle$ - and $[[\cdot]]$ -components in Γ). Each agent will have r available actions $\{0, \dots, r-1\}$ at the root state s , hence $\{0, \dots, r-1\}^k$ 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 Γ for the satisfaction of which the agent chooses to act. For every such action profile σ we denote by $N(\sigma)$ the set of agents $\{i \mid \sigma_i \geq m\}$ and then we define the number $\mathbf{neg}(\sigma)$ 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 $\mathbf{neg}(\sigma)$ between 0 and $n-1$ they wish, i.e., to set the "collective action" of all agents in $\mathbf{neg}(\sigma)$ on any $[[\cdot]] X$ -formula in Γ they choose.) Now, we consider the set

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

Note that Δ_σ is a subset of Γ of distributed control if it contains a formula

$[[B_j]]X\Psi$, or else can be made a set of distributed control by adding $[[\mathbb{A}]]X\top$ to it. Indeed, all agents in a A_j choose j , so all coalitions A_j must be pairwise disjoint. Besides, if $[[B_l]]X\Psi_l \in \Delta_\sigma$ then it is clearly a unique $[[\cdot]]$ -formula in Δ_σ and no agents from any $A_j \in \Delta_\sigma$ are in $N(\sigma)$, hence $A_j \subseteq B_l$ for each $\langle\langle A_j \rangle\rangle X\Phi_j \in \Delta_\sigma$. Thus, Δ_σ is one of $\Delta_1, \dots, \Delta_d$, say Δ_i . Then, we determine the successor state $\text{out}(s, \sigma)$ to be s_i . To complete the definition of the CGM, at each successor state s_i of s we graft a copy of \mathcal{M}_i .

We will show that the resulting CGM \mathcal{M} satisfies Γ at s . Indeed, for every $\langle\langle A_j \rangle\rangle X\Phi_j \in \Gamma$ 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 \mathcal{M}_i the satisfaction of the objective Φ_j if the play enters the copy of \mathcal{M}_i , and acting in an arbitrarily fixed manner at all other states of \mathcal{M} . (Note that, if the strategy for A_j in \mathcal{M}_i is positional, then the above described strategy is positional, too.) Lastly, every $[[B_l]]X\Psi_l \in \Gamma$ is true at s , too, because if $B_l \neq \mathbb{A}$ then for every collective action of all agents from $[[B_l]]X$ there is a suitable complementary action of $\mathbb{A} \setminus B_l$, where all agents choose actions greater than m and such that $\text{neg}(\sigma)$ adds up to l modulo n . (In fact, this can be guaranteed by any agent in $\mathbb{A} \setminus B_l$ after all others have chosen their actions.) In the case when $B_l = \mathbb{A}$, every subset $\{\langle\langle A_j \rangle\rangle X\Phi_j, [[B_l]]X\Psi_l\}$ for $j < m$ is of distributed control, hence Ψ_l is true at the root s_i of \mathcal{M}_i for each $i = 1, \dots, d$. Thus, $\mathcal{M}, s \models \Gamma$, which completes the proof. \square

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 $\text{St}(\text{ATL}_1^*)$ 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 $\text{St}(\text{ATL}_1^*)$ 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 $\langle\langle A \rangle\rangle\theta$ 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. \square

5 Optimal decision procedures for satisfiability in fragments of ATL_1^*

5.1 Centipede models. Satisfiability in LTL_1 , CTL_1 and CTL_1^+

Satisfiability of LTL_1 is analyzed in [5]. In particular, it is shown that if an LTL_1 formula θ is satisfiable then it is satisfiable in a model of the form $s_0 s_1 \dots s_\ell^\omega$ where $\ell = |\theta|$. Consequently, it is shown that satisfiability of LTL_1 is NP-complete. We provide similar results for CTL_1^+ and CTL_1 here.

Proposition 5.1 *If a CTL_1^+ formula φ has a model, then it has a model with at most $O(|\varphi|^2)$ states.*

Proof. Suppose $\mathcal{M}, s_0 \models \varphi$ for a CTL_1^+ formula φ , a model \mathcal{M} and a state s_0 . Assume w.l.o.g. that φ is in NNF. We generate another model \mathcal{M}' with $O(|\varphi|^2)$ states and a state s'_0 such that $\mathcal{M}', s'_0 \models \varphi$. Let Δ_Q be the set of subformulae of φ that has Q as main connective for $Q \in \{\text{E}, \text{A}\}$ and let Δ_B be the set of maximal Boolean subformulae of φ that do not occur in the scope of a path quantifier. For each $Z \in \{\text{E}, \text{A}, \text{B}\}$ let $\Delta_Z^\top \subseteq \Delta_Z$ be the subsets satisfied in \mathcal{M}, s_0 . Now, for each $\text{E}\psi \in \Delta_E^\top$ let $\rho^\psi = \rho_0^\psi \rho_1^\psi \dots$ be a path in \mathcal{M} starting in s_0 such that $\rho^\psi \models \psi$. Since $\mathcal{M}, s_0 \models \varphi$ we have for every $\text{A}\psi' \in \Delta_A^\top$ that $\rho^\psi \models \psi'$ because ψ' is satisfied along all paths from s_0 . Further, $\rho_0^\psi = s_0$ implies that $\rho^\psi \models \psi \wedge \bigwedge_{\psi' \in \Delta_A^\top} \psi' \wedge \bigwedge_{\beta \in \Delta_B^\top} \beta$. Since this is an LTL_1 formula of size at most $|\varphi|$ it has a model π^ψ of the form $\pi_0^\psi \dots (\pi_{|\varphi|}^\psi)^\omega$ where π_0^ψ is labelled as s_0 . Now, by gluing together each path π^ψ (which is made finite by adding a self-loop to the state $\pi_{|\varphi|}^\psi$) in the initial state s'_0 we obtain a transition system \mathcal{M}' such that $\mathcal{M}', s'_0 \models \varphi$. Since $|\Delta_E^\top| \leq |\varphi|$ there are at most $O(|\varphi|^2)$ states in \mathcal{M}' . \square

Further, we will see that for satisfiable formulae of CTL_1^* , ATL_1 and $\text{St}(\text{ATL}_1^*)$ 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.

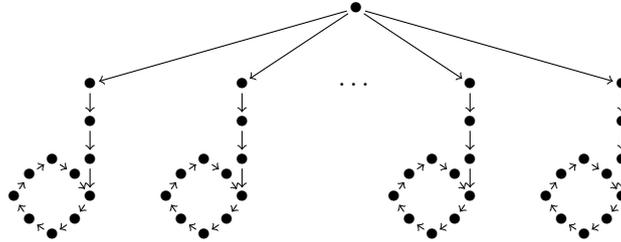


Fig. 2. A centipede model

However, for the flat fragments CTL_1^* , ATL_1 , $\text{St}(\text{ATL}_1^*)$ models of polynomial size are not guaranteed to exist as for CTL_1^+ . 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_1^* and $\text{St}(\text{ATL}_1^*)$. Second, in the cases of ATL_1 and $\text{St}(\text{ATL}_1^*)$ (but not for CTL_1^*) an exponential branching factor in the initial state may be forced by a formula. Indeed, consider the following ATL_1 formula

$$\varphi = \bigwedge_{i=1}^n \langle\langle i \rangle\rangle X p_i \wedge \langle\langle i \rangle\rangle X \neg p_i$$

over the propositions $\{p_1, \dots, p_n\}$ and players $\{1, \dots, n\}$. For a state s_0 to satisfy this formula there has to be a successor state for each possible truth assignment

to the propositions $\{p_1, \dots, p_n\}$, of which there are 2^n . This phenomenon does not occur in the branching-time logic CTL_1^* .

Proposition 5.2 *Satisfiability in CTL_1^+ is NP-complete.*

Proof. NP-hardness follows directly from Boolean satisfiability. An NP-algorithm for CTL_1^+ works as follows. It takes as input a CTL_1^+ formula φ in NNF, hence a positive Boolean combination of flat CTL_1^+ state formulae, and guesses non-deterministically a centipede model \mathcal{M}, s_0 of size $O(|\varphi|^2)$ for φ , as well as the disjuncts in φ that evaluate to true at s_0 . (According to Proposition 5.1, if φ has a model then it has a model of this form and size.) After guessing, it checks whether the resulting formula of the form $\varphi' = \beta \wedge \bigwedge_{i=0}^{\ell} \varphi'_i$ is true in the guessed model where β is a Boolean formula and each φ'_i is of the form $A\theta_i$ or $E\theta_i$ for an LTL_1 formula θ_i . First, the model-checking of β can be done in linear time. Next, for each of the $O(|\varphi|)$ formulae θ_i it can be checked whether it is true in each of the $O(|\varphi|)$ paths of the centipede model in polynomial time since 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 CTL_1 is NP-complete.*

5.2 Lower bound for satisfiability in ATL_1

Proposition 5.4 *$\text{ATL}_1\text{-SAT}$ is Σ_3^P -hard.*

Proof. The proof is by reduction from the Σ_3^P -SAT problem, which is Σ_3^P -complete. This problem takes as input a quantified Boolean sentence

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

where γ' is a Boolean formula over the Boolean variables x_1, \dots, x_n . The output is true if and only if γ is true. Given γ , we construct an ATL_1 formula $\psi(\gamma)$ over the set $\text{Prop} = \{x_1, \dots, x_n\}$ of proposition symbols as follows

$$\psi = \bigwedge_{j=1}^m (\text{AX } x_j \vee \text{AX } \neg x_j) \wedge \bigwedge_{i=m+1}^n (\langle\langle i \rangle\rangle \text{X } x_i \wedge \langle\langle i \rangle\rangle \text{X } \neg x_i) \wedge \neg \langle\langle m+1, \dots, k \rangle\rangle \text{X } \neg \gamma'$$

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

First, suppose that γ is true. Then we construct a CGM $\mathcal{M} = (\mathbb{A}, \text{St}, \{\text{Act}_a\}_{a \in \mathbb{A}}, \{\text{act}_a\}_{a \in \mathbb{A}}, \text{out}, \text{Prop}, \text{L})$ and a state $s_0 \in \text{St}$, such that $\mathcal{M}, s_0 \models \psi(\gamma)$, as follows. Let $\mathbb{A} = \{m+1, \dots, n\}$, $\text{St} = \{s_0\} \cup \{s_{v_{m+1}, \dots, v_n} \mid v_i \in \{0, 1\} \text{ for } m+1 \leq i \leq n\}$, $\text{Act}_a = \{0, 1\}$ for all $a \in \mathbb{A}$. Then, for every agent $a \in \mathbb{A}$ 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 $\text{out}(s_0, \langle v_{m+1}, \dots, v_n \rangle) = s_{v_{m+1}, \dots, v_n}$ and $\text{out}(s, \alpha_{\mathbb{A}}) = s$ for all $s \neq s_0$ and all action profiles $\alpha_{\mathbb{A}}$. The set of proposition symbols is $\text{Prop} = \{x_1, \dots, x_n\}$. The labelling is given by $L(s_0) = \emptyset$ and for every

$i \in \{m+1, \dots, n\}$ we have $x_i \in L(s_{v_{m+1}, \dots, v_n})$ if and only if $v_i = 1$ when $s_{v_{m+1}, \dots, v_n} \in \text{St} \setminus \{s_0\}$. Finally, let x'_1, \dots, x'_m be particular values such that $\forall x_{m+1}, \dots, x_k \exists x_{k+1}, \dots, x_n. \gamma'[x_1 \mapsto x'_1, \dots, x_m \mapsto x'_m]$ is true. Such values exist since γ is true. For $1 \leq i \leq m$ and every $s_{v_{m+1}, \dots, v_n} \in \text{St} \setminus \{s_0\}$, let $x_i \in L(s_{v_{m+1}, \dots, v_n})$ 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, \dots, x'_m . The subformula $\bigwedge_{i=m+1}^n (\langle\{i\}\rangle X x_i \wedge \langle\{i\}\rangle X \neg x_i)$ is clearly true at s_0 . The same is the case for $\bigwedge_{j=1}^m (\text{AX } x_j \vee \text{AX } \neg x_j)$. Next, since γ 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 $\{m+1, \dots, k\}$ there exists values of x_i for players in $\{k+1, \dots, n\}$ such that γ' is true in the successor state. Thus, coalition $\{m+1, \dots, k\}$ does not have a strategy to ensure that γ' 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 γ is false. Then for all x_1, \dots, x_m there exists x_{m+1}, \dots, x_k such that γ' is false for all x_{k+1}, \dots, 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 $\bigwedge_{j=1}^m (\text{AX } x_j \vee \text{AX } \neg x_j)$. In this case there exists particular values x'_i for $m+1 \leq i \leq k$ such that γ' is false for all x_{k+1}, \dots, x_n when x_i take the values x'_i for $m+1 \leq i \leq k$. Consider the strategy for coalition $\{m+1, \dots, k\}$ that chooses these values for x_i in the successor state for $m+1 \leq i \leq k$. This strategy ensures that γ' is false in the successor state. However, this contradicts the fact that $\mathcal{M}, s_0 \models \neg \langle\{m+1, \dots, k\}\rangle X \neg \gamma'$. Thus, γ 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 ATL_1^+ in the following section. Thus, the $\langle\cdot\rangle X$ fragment of ATL_1 is as hard as the full ATL_1^+ .

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

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

$$\delta(v) := \bigwedge_{v(p_i)=\top} p_i \wedge \bigwedge_{v(p_i)=\perp} \neg p_i$$

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

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, \dots, p_r . Then, Φ 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 PSPACE.*

(ii) *The satisfiability testing for ATL_1^+ (and ATL_1) is in Σ_3^P .*

Proof. The decision procedures for both $\text{St}(\text{ATL}_1^*)$ and ATL_1^+ will be essentially the same, but in their last phases they work in different computational complexities. First, consider an $\text{St}(\text{ATL}_1^*)$ formula Φ and let $\text{Prop}(\Phi) = \{p_1, \dots, p_r\}$. The formula Φ is a Boolean combination of atomic propositions and subformulae of the type $\langle\langle C \rangle\rangle\theta$ where $\theta \in \text{LTL}$. By Lemma 4.6, we can assume that each such θ is in a LSNF of linearly increased length, i.e., is a Boolean combination of atomic propositions and X-formulae (formulae beginning with X) of LTL. The algorithm now works as follows:

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

2. After elementary Boolean simplifications (of the type $\top \wedge A \equiv A$, $\perp \wedge A \equiv \perp$, etc.) each $v[\theta]$ is transformed to a Boolean combination of X-formulae only. Using the LTL validities $X\eta \wedge X\zeta \equiv X(\eta \wedge \zeta)$ and $X\eta \vee X\zeta \equiv X(\eta \vee \zeta)$, it is further equivalently transformed into an X-formula which is at most as long.

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

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

$$\Theta = \langle\langle A_0 \rangle\rangle X\theta_0 \wedge \dots \wedge \langle\langle A_{m-1} \rangle\rangle X\theta_{m-1} \wedge [[B_0]]X\eta_0 \wedge \dots \wedge [[B_{n-1}]]X\eta_{n-1}$$

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

$$D(\Theta) = \{\langle\langle A_0 \rangle\rangle X\theta_0, \dots, \langle\langle A_{m-1} \rangle\rangle X\theta_{m-1}, [[B_0]]X\eta_0, \dots, [[B_{n-1}]]X\eta_{n-1}, [[A]]X\top\}$$

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 LTL formulae, these checks can be done using standard techniques.

Each check in step 4. of the algorithm can be done in PSPACE when Φ is a $\text{St}(\text{ATL}_1^*)$ formula, since each successor component is an LTL formula. In the case of ATL_1^+ the checks can be done in NP according to [5], as in this case each successor component is an 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 coNP^{NP} for 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 ATL_1^+ , by guessing the true propositions in the initial state and the true disjuncts in Φ , and then applying resp. a PSPACE-oracle and coNP^{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) CTL_1^* *is PSPACE-complete*
- (iii) ATL_1^+ *is Σ_3^P -complete*
- (iv) ATL_1 *is Σ_3^P -complete*

Here is another consequence of the proof of Proposition 5.6:

Corollary 5.8 *Every satisfiable $\text{St}(\text{ATL}_1^*)$ formula Φ has a centipede model \mathcal{M} with branching factor $O(2^{|\Phi|})$ in the root. Further, every ultimately periodic path in \mathcal{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 ATL_1^*

The decision procedure for $\text{St}(\text{ATL}_1^*)$ can be extended to a PSPACE-complete decision procedure for the whole ATL_1^* , by combining it with a PSPACE decision procedure for LTL and showing that every path-satisfiable 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 $\widehat{\text{ATL}}_1^*$, 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 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 $\widehat{\text{ATL}}_1^*$ formula Φ is satisfied in a LoC model with size bounded exponentially in $|\Phi|$.*

Proof. Given an $\widehat{\text{ATL}}_1^*$ formula Φ we define its LTL *skeleton* $\text{Sk}_{\text{LTL}}(\Phi)$ as follows: Let the state subformulae of Φ of type $\langle\langle C \rangle\rangle\theta$ or $[[C]]\theta$ be Ψ_1, \dots, Ψ_n . For each of them Ψ we introduce a new (not in **Prop**) atomic proposition p_Ψ . Then we produce the LTL formula $\widehat{\Phi}$ by replacing every occurrence of such a subformula Ψ in Φ by p_Ψ . Now, define

$$\text{Sk}_{\text{LTL}}(\Phi) ::= \widehat{\Phi} \wedge \bigwedge_i^n G(p_{\Psi_i} \rightarrow \Psi_i)$$

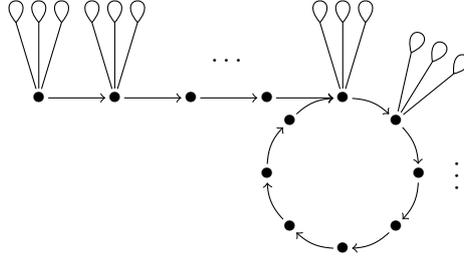


Fig. 3. A Lasso of Centipedes (LoC) model

We claim that any CGM \mathcal{M} and a path π in it on which Φ is true can be expanded to a CGM $\tilde{\mathcal{M}}$ and a path $\hat{\pi}$ in it satisfying $\text{Sk}_{\text{LTL}}(\Phi)$, by evaluating each new atomic proposition p_Ψ to be true at exactly those states of π at which Ψ is true in \mathcal{M} . Conversely, for any CGM \mathcal{M} and a path π in it on which $\text{Sk}_{\text{LTL}}(\Phi)$ is true, the formula Φ is true on π , too, because all atomic propositions p_{Ψ_i} occur only positively in $\hat{\Phi}$, so replacing them with the respective Ψ_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 \mathcal{M} and a path π in it on which $\text{Sk}_{\text{LTL}}(\Phi)$ is true. Then, in particular, the path π 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 π 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 π . Now, for every state \hat{s} on $\hat{\pi}$, let s be its prototype in π . We do the following.

- Consider the set $\Gamma(s)$ of state subformulae Ψ of Φ such that p_Ψ is in the label of s in π . Since $\text{Sk}_{\text{LTL}}(\Phi)$ is true on π , every formula in $\Gamma(s)$ is true at s in \mathcal{M} . Thus, $\Gamma(s)$ is satisfiable, hence by Corollary 5.8, it can be satisfied at the root state of a centipede model $\mathcal{M}(\Gamma(s))$ of exponentially bounded in $|\Phi|$ number and length of legs.
- Now, we graft a copy of $\mathcal{M}(\Gamma(s))$ 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 $\mathcal{M}(\Gamma(s))$, 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 $\tilde{\mathcal{M}}$ 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, $\tilde{\mathcal{M}}, \hat{\pi} \models \text{Sk}_{\text{LTL}}(\Phi)$, hence $\tilde{\mathcal{M}}, \hat{\pi} \models \Phi$. \square

For lack of space we only briefly indicate the additional complication in extending this result to ATL_1^* : if a subformula $\Psi = \langle\langle\emptyset\rangle\rangle\theta$ is true at some state of the path π in the CGM satisfying Φ , its effect cannot be constrained only on the centipede model grafted at the respective state of $\widehat{\mathcal{M}}$, 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 π 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 ATL_1^* 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 θ [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 Δ (of at most polynomial size) of eventuality formulae is kept to make sure that all eventualities that are needed for θ to be true are actually true further on the lasso.

The algorithm for ATL_1^* works in the same way on an ATL_1^* formula Φ , but treats strategically quantified subformulae of Φ 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 $\text{St}(\text{ATL}_1^*)$ 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 θ then the play goes into the centipede (and stays there). But, when $A = \emptyset$ then θ 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 Δ of formulae we are keeping only needs to be satisfied around the lasso. To keep track of this we keep, in addition to Δ , an extra set of formulae Γ which must be satisfied both around the lasso and on paths that exits to a centipede. Thus, the formulae in Γ must be included in the $\text{St}(\text{ATL}_1^*)$ satisfiability check at each step. But since Γ 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. \square

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

6 Concluding remarks and summary of results

We have developed optimal decision procedures for the satisfiability problems in flat fragments of ATL^* , and in particular CTL^* 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.

\mathcal{L}	SAT(\mathcal{L})	SAT(\mathcal{L}_1)
LTL	PSPACE [14]	NP [5]
CTL	EXPTIME [6]	NP (Cor. 5.3)
CTL ⁺	2EXPTIME [10]	NP (Prop. 5.2)
CTL*	2EXPTIME [6]	PSPACE (Theo. 5.7)
ATL	EXPTIME [16]	Σ_3^P (Theo. 5.7)
ATL ⁺	2EXPTIME [12][10]	Σ_3^P (Theo. 5.7)
ATL*	2EXPTIME [12]	PSPACE (Theo. 5.7, Cor. 5.11)

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

References

- [1] Ågotnes, T., V. Goranko and W. Jamroga, *Alternating-time temporal logics with irrevocable strategies*, in: D. Samet, editor, *Proc. of TARK XI* (2007), pp. 15–24.
- [2] Alur, R., T. A. Henzinger and O. Kuperman, *Alternating-time temporal logic*, *Journal of the ACM* **49** (2002), pp. 672–713.
- [3] Brihaye, T., A. Da Costa, F. Laroussinie and N. Markey, *ATL with strategy contexts and bounded memory*, in: *Proc. of LFCS'2009, Springer LNCS 5407*, 2009, pp. 92–106.
- [4] Bulling, N. and W. Jamroga, *Comparing variants of strategic ability: how uncertainty and memory influence general properties of games*, *J. of AAMAS* (2013), pp. 1–45.
- [5] Demri, S. and P. Schnoebelen, *The complexity of propositional linear temporal logics in simple cases*, *Inf. Comput.* **174** (2002), pp. 84–103.
- [6] Emerson, E. A., *Temporal and modal logic*, in: *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics (B)*, 1990 pp. 995–1072.
- [7] Goranko, V. and D. Shkatov, *Tableau-based decision procedures for logics of strategic ability in multiagent systems*, *ACM Trans. Comput. Log.* **11** (2009).
- [8] Goranko, V. and G. van Drimmelen, *Complete axiomatization and decidability of Alternating-time temporal logic*, *Theor. Comp. Sci.* **353** (2006), pp. 93–117.
- [9] Halpern, J. Y., *The effect of bounding the number of primitive propositions and the depth of nesting on the complexity of modal logic*, *Artif. Intell.* **75** (1995), pp. 361–372.
- [10] Johannsen, J. and M. Lange, *CTL⁺ is Complete for Double Exponential Time*, in: *Proc. ICALP'03* (2003), pp. 767–775.
- [11] Markey, N. and P. Schnoebelen, *Model checking a path*, in: *CONCUR*, 2003, pp. 248–262.
- [12] Schewe, S., *ATL* satisfiability is 2EXPTIME-complete*, in: *Proc. of ICALP (2)*, 2008, pp. 373–385.
- [13] Schröder, L. and Y. Venema, *Flat coalgebraic fixed point logics*, in: *Proc. of CONCUR'2010* (2010), pp. 524–538.
- [14] Sistla, A. P. and E. M. Clarke, *The complexity of propositional linear temporal logics*, *J. ACM* **32** (1985), pp. 733–749.
- [15] Troquard, N. and D. Walther, *On satisfiability of ATL with strategy contexts*, in: *Proc. of JELIA '12*, *LNAI 7519* (2012), pp. 398–410.
- [16] van Drimmelen, G., *Satisfiability in alternating-time temporal logic*, in: *Proc. of LICS'03*, 2003, pp. 208–217.
- [17] Walther, D., C. Lutz, F. Wolter and M. Wooldridge, *ATL satisfiability is indeed ExpTime-complete*, *Journal of Logic and Computation* **16** (2006), pp. 765–787.