Results on alternating-time temporal logics with linear past

Laura Bozzelli, Aniello Murano, and Loredana Sorrentino

University of Napoli "Federico II", Italy

Abstract

We investigate the succinctness gap between two known equally-expressive and different linear-past extensions of standard CTL^* (resp., ATL^*). We establish by formal non-trivial arguments that the 'memoryful' linear-past extension (the history leading to the current state is taken into account) can be exponentially more succinct than the standard 'local' linear-past extension (the history leading to the current state is forgotten). As a second contribution, we consider the ATL-like fragment, denoted ATL_{lp}, of the known 'memoryful' linear-past extension of ATL^*. We show that ATL_{lp} is strictly more expressive than ATL, and interestingly, it can be exponentially more succinct than the more expressive logic ATL^*. Moreover, we prove that both satisfiability and model-checking for the logic ATL_{lp} are EXPTIME-complete.

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

Temporal logics provide a fundamental framework for the description of the dynamic behavior of reactive systems. Additionally, they support the successful model-checking approach [3] that allow complex (finite-state) systems, usually modeled by propositional Kripke structures, to be verified automatically. The model-checking methodology considers three types of temporal logics which differ in the underlying nature of time: linear, branching, and alternating. In linear-time temporal logics such as standard LTL [27], formulas are interpreted over linear sequences (corresponding to single paths of the Kripke structure), and temporal operators are provided for describing the ordering of events along a single computation path. Branching-time temporal logics such as CTL [10] and CTL^* [11], on the other hand, allow to reason about several possible futures: formulas are interpreted over states of a Kripke structure, hence referring to all the possible system computations. Such logics are in general more expressive than linear-time temporal logics since they provide both temporal operators for describing properties of a single path, and path quantifiers for describing the branching structure in computation trees (resulting from unwinding a Kripke structure from the initial state). Finally, alternating-time temporal logic such as ATL^* and ATL [2], generalize the branching-time paradigm (useful for the verification of closed systems) to a strategic-reasoning paradigm suitable for the verification of open and multi-agent systems [20, 2, 1, 6, 9, 25, 26, 15, 5]. In this setting, different processes or components (the agents) can interact in an adversarial or cooperative manner in order to achieve given temporal goals. The interaction among agents is usually modeled by *concurrent game structures* [2] (CGS, for short), a generalization of Kripke structures, where each transition results from a set of decisions, one for each agent. In particular, the logic ATL^*, which is interpreted over CGS, is an extension of CTL^* obtained by replacing path quantifiers with more general quantifiers $\langle\langle A\rangle\rangle$ parameterized by a set A of agents which express selective quantification over those paths (from the current state) that are obtained as outcomes of the infinite game between the coalition A and the complement.

Linear past in temporal logics. Standard temporal logics such as LTL, CTL^*, and ATL^* do not have explicit mechanisms to refer to the past of the current time. On the other

Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

hand, it is well-known that temporal logics which combine both past and future temporal modalities make specifications easier to write and more natural. In particular, the past extension LTL_p of standard LTL does not increase the complexity of model-checking and satisfiability-checking [32], and at the same time, turns out to be exponentially more succinct than LTL [22]. For branching-time temporal logics, the adding of past-time constructs has been investigated in many papers [30, 14, 17, 23, 33, 24, 4, 18]. Usually, the past is assumed to be finite (since program computations have a definite starting time) and cumulative (i.e., the history of the current situation increases with time and is never forgotten). Moreover, one can adopt either a branching-past approach (past and future are handled uniformly) or a linear-past approach. Here, we focus on known linear-past extensions of CTL^* and its alternating-time counterpart ATL^*, which are syntactically obtained by adding the past versions of the standard LTL temporal modalities. The simplest linear-past extension of CTL^* is the logic PCTL^* [14], where the semantics of path quantification is the same as for CTL^*: path quantification ranges over paths that starts in the current node of the computation tree. Hence, the history (computation) from the root-node (starting time) to the current node is forgotten (*local linear-past view*). A similar local linear-past extension, denoted PATL^*, can be considered for the logic ATL^*, where the linear-time temporal goals, arguments of the strategy quantifiers, are evaluated along the outcomes of the selected strategy starting from the current node. A more interesting and meaningful linear-past extension of CTL^* is the logic CTL^*_{lp} [17, 18], where path quantification is 'memoryful', i.e., it ranges over paths that start at the root and visit the current node (*memoryful linear-past view*). The ATL^* counterpart of the logic CTL^*_{lp}, here denoted by ATL^*_{lp}, has been investigated in [26] (see also [7]), where the temporal goals are evaluated at the current time of the paths obtained by prefixing the outcomes of the selected strategy with the history leading to the current node. The usefulness of memoryful linear-past has been illustrated in [18, 26]: for example, one can require that a condition is satisfied only if a precondition holds along the whole past computation. In strategic reasoning, as argued in [26], memoryful linear-past enables *relentful* reasoning (agents can relent and change their goals depending on the history) which can be applied to relevant scenarios such as strong cyclic planning and bounded verification. Satisfiability and model-checking of the mentioned linear-past extensions of CTL^* (resp., ATL^*) have the same complexity as CTL^* (resp., ATL^*) with the exception of model-checking against CTL^*_{lp} which is EXPSPACE-complete [18], hence, exponentially harder than CTL^* model-checking (the latter being PSPACE-complete [11]).

Our contribution. It is known that PCTL^* and CTL^*_{lp} have the same expressiveness as CTL^* [18] and there are translations from PCTL^* and CTL^*_{lp} into CTL^* of non-elementary complexity [18] based on the separation theorem for LTL_p [12]. On the other hand, the ability to refer to the past makes both PCTL^* and CTL^*_{lp} exponentially more succinct than CTL^*. Analogous results hold for the logics PATL^* and ATL^*_{lp} when compared to ATL^* [26]. On the other hand, no succinctness gap is known between the 'memoryful' past view in CTL^*_{lp} (resp., ATL^*_{lp}) and the 'local' past view in PCTL^* (resp., PATL^*). Our first contribution addresses this issue: we establish by formal non-trivial arguments that CTL^*_{lp} (resp., ATL^*_{lp}) can be exponentially more succinct than PCTL^* (PATL^*).

The logics CTL [11] and ATL [2] are well-known syntactical fragments of CTL^* and ATL^* which have received a lot of attention due to the existence of polynomial-time algorithms which solve the associated model-checking problem. As a second contribution, we investigate the ATL-like fragment ATL_{lp} of ATL^*_{lp} which, to the best of our knowledge, has not been considered so far in the literature. In fact, a past perfect recall extension of ATL with a semantics equivalent to that of ATL_{lp} has been studied in [13] under an *imperfect information* *setting*, and in particular, for such a setting, a model-checking algorithm of non-elementary complexity is provided [13]. We show that ATL_{lp} is strictly more expressive than ATL, and interestingly, and perhaps surprisingly, it can be exponentially more succinct than the more expressive logic ATL^*. Moreover, we establish that both satisfiability and model-checking for the logic ATL_{lp} are EXPTIME-complete. Hence, while ATL_{lp} satisfiability has the same complexity as ATL satisfiability, model-checking against ATL_{lp} is exponentially harder than ATL model-checking. The upper bounds are obtained by an automata-theoretic framework based on the use of a subclass of Büchi alternating automata for CGS (Büchi ACG) [29], called *ACG with satellites* and introduced in [26]. Due to lack of space, some proofs are omitted and can be found in the Appendix.

2 Preliminaries

We fix the following notations. Let AP be a finite non-empty set of atomic propositions, Ag be a finite non-empty set of agents, and Ac be a finite non-empty set of actions that can be taken by agents. Given a set $A \subseteq Ag$ of agents, an A-decision d_A is an element in Ac^A assigning to each agent $ag \in A$ an action $d_A(ag)$. For $A, A' \subseteq Ag$ such that $A \cap A' = \emptyset$, an A-decision d_A and A'-decision $d_{A'}$, $d_A \cup d_{A'}$ denotes the $(A \cup A')$-decision defined in the obvious way. Let $Dc = Ac^{Ag}$ be the set of *full decisions* of all the agents in Ag.

Let $\mathbb{N}$ be the set of natural numbers. For all $i, j \in \mathbb{N}$, with $i \leq j$, $[i, j]$ denotes the set of natural numbers h such that $i \leq h \leq j$. Let w be a finite or infinite word over some alphabet Σ. By $|w|$ we denote the length of w (we write $|w| = \infty$ if w is infinite). For all $i, j \in \mathbb{N}$, with $i \leq j < |w|$, $w(i)$ denotes the i^{th} letter of w, and $w[i, j]$ the infix of w between positions i and j, i.e., the finite word $w(i)w(i+1)\ldots w(j)$.

Given a set Υ of directions, an (*infinite*) Υ-*tree* T is a prefix closed subset of Υ^* such that for all $x \in T$, $x \cdot \gamma \in T$ for some $\gamma \in \Upsilon$. Elements of T are called nodes and ε is the root of T. For $x \in T$, the set of children of x in T is the set of nodes of the form $x \cdot \gamma$ for some $\gamma \in \Upsilon$. A path of T is a non-empty finite or infinite sequence π of nodes such that $\pi(i)$ is a child in T of $\pi(i-1)$ for all $0 < i < |\pi|$. A path π of T is *initial* if it starts at the root, i.e. $\pi(0) = \varepsilon$.

For an alphabet Σ, a Σ-labeled Υ-tree is a pair $\langle T, Lab\rangle$ consisting of a Υ-tree and a labelling $Lab : T \mapsto \Sigma$ assigning to each node in T a symbol in Σ. We extend the labeling Lab to paths in the obvious way, i.e. $Lab(\pi)$ denotes the word over Σ of length $|\pi|$ given by $Lab(\pi(0))Lab(\pi(1))\ldots$. W.l.o.g. we focus on labeled trees with a finite branching degree (i.e., each node has a finite set of children).

2.1 Concurrent Game Structures

Concurrent game structures (CGS, for short) [2] generalize Kripke structures to a setting with multiple agents (or players). They can be viewed as multi-player arenas in which players perform concurrent actions, chosen strategically as a function of the history of the game.

▶ **Definition 1** (CGS). A CGS (over AP, Ag, and Ac) is a tuple $\mathcal{G} = \langle S, s_0, Lab, \tau\rangle$, where S is a set of states, $s_0 \in S$ is the initial state, $Lab : S \mapsto 2^{AP}$ maps each state to a set of atomic propositions, and $\tau : S \times Dc \mapsto S$ is a transition function that maps a state and a full decision to a target state. The CGS $\mathcal{G}$ is finite if S is finite. A *state s is controlled by an agent ag* if for all $\{ag\}$-decisions d and counter-decisions $d_1, d_2 \in Ac^{Ag\setminus\{ag\}}$, $\tau(s, d \cup d_1) = \tau(s, d \cup d_2)$.

We now recall the notion of *strategy* and *counter strategy* in a CGS $\mathcal{G} = \langle S, s_0, Lab, \tau\rangle$. For a state s, the set of successors of s is the set of states s' such that $s' = \tau(s, d)$ for some full decision d. A *play* is an infinite sequence of states $s_1 s_2 \ldots$ such that s_{i+1} is a successor of s_i

for all $i \geq 1$. An *history* (or *track*) ν is a non-empty prefix of some play. We denote by $lst(\nu)$ the last state of ν. Let Trk be the set of tracks in $\mathcal{G}$. Given a set $A \subseteq Ag$ of agents, a *strategy for* A is a mapping $f_A : Trk \mapsto Ac^A$ assigning to each track ν an A-decision. For a track ν, the set $out(\nu, f_A)$ of *plays consistent with* f_A *and* ν (also called *outcomes of* f_A *from* ν) is the non-empty set of plays of the form $\pi = \nu' \cdot s_1 s_2 \ldots$ such that $\nu' \cdot s_1 = \nu$ and for all $i \geq 1$, there is a decision $d \in Ac^{Ag \setminus A}$ of agents in $Ag \setminus A$ such that $s_{i+1} = \tau(s_i, f_A(\nu' \cdot s_1 \ldots s_i) \cup d)$.

Thus, the *outcome* function $out(\nu, f_A)$ returns the set of all the plays having ν as prefix that can occur when agents A execute strategy f_A from the history ν on.

A *counter strategy* is a mapping $f_A^c : Trk \times Ac^A \mapsto Ac^{Ag \setminus A}$ from tracks and decisions of the agents in A to decisions of the agents in $Ag \setminus A$. For a track ν, the set $out(\nu, f_A^c)$ of *plays consistent with the counter strategy* f_A^c *and* ν is the non-empty set of plays of the form $\pi = \nu' \cdot s_1 s_2 \ldots$ such that $\nu' \cdot s_1 = \nu$ and for all $i \geq 1$, there is a decision $d \in Ac^A$ of agents in A so that $s_{i+1} = \tau(s_i, f_A^c(\nu' \cdot s_1 \ldots s_i, d) \cup d)$. For a state s, an *s-play* is a play starting from state s. We are interested in the computation trees induced by CGS.

▶ **Definition 2.** For a set Υ of directions, a *Concurrent Game* Υ*-Tree* (Υ-CGT) is a CGS $\langle T, \varepsilon, Lab, \tau \rangle$, where $\langle T, Lab \rangle$ is a 2^{AP}-labeled Υ-tree, and for each node $x \in T$, the set of successors of x corresponds to the set of children of x in T. Every CGS $\mathcal{G} = \langle S, s_0, Lab, \tau \rangle$ induces a S-CGT, called *computation tree of* $\mathcal{G}$ and denoted by $Unw(\mathcal{G})$, obtained by unwinding $\mathcal{G}$ from the initial state in the usual way. Formally, $Unw(\mathcal{G}) = \langle T, \varepsilon, Lab', \tau' \rangle$, where T is the set of elements ν in S^* such that $s_0 \cdot \nu$ is a track of $\mathcal{G}$, and for all $\nu \in T$ and $d \in Dc$, $Lab'(\nu) = Lab(lst(\nu))$ and $\tau'(\nu, d) = \nu \cdot \tau(lst(\nu), d)$, where $lst(\varepsilon) = s_0$.

A *Concurrent Game Tree* (CGT, for short) is a Υ-CGT for some set Υ of directions.

2.2 Alternating-time temporal logic with linear past

In this section, we recall the 'memoryful' linear-past extension of alternating-time temporal logic ATL*, introduced in [26] and, here, denoted by ATL$^*_{lp}$.

For the given sets AP and Ag of atomic propositions and agents, the set of ATL$^*_{lp}$ *path* formulas φ are defined by the following grammar:

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}\varphi \mid \mathsf{X}^-\varphi \mid \varphi \,\mathsf{U}\, \varphi \mid \varphi \,\mathsf{U}^-\, \varphi \mid \langle\langle A \rangle\rangle\varphi$$

where $p \in AP$, $A \subseteq Ag$, X and U are the standard "next" and "until" linear temporal modalities, X^- ("previous") and U^- ("since") are their past counterparts, respectively, and $\langle\langle A \rangle\rangle$ is the "existential strategic quantifier" parameterized by a set of agents. Formula $\langle\langle A \rangle\rangle\varphi$ expresses that the group of agents A has a collective strategy to enforce the temporal property φ. We also use some shorthands:

- the *eventually* modality $\mathsf{F}\varphi := \top \,\mathsf{U}\, \varphi$ and its past counterpart $\mathsf{F}^-\varphi := \top \,\mathsf{U}^-\, \varphi$,
- the *always* modality $\mathsf{G}\varphi := \neg\mathsf{F}\neg\varphi$ and its past counterpart $\mathsf{G}^-\varphi := \neg\mathsf{F}^-\neg\varphi$,
- the *release* modality $\varphi_1 \mathsf{R} \varphi_2 := \neg(\neg\varphi_1 \mathsf{U} \neg\varphi_2)$.

A *state formula* is a formula where each temporal modality is in the scope of a strategic quantifier. A *basic formula* is a state formula of the form $\langle\langle A \rangle\rangle\varphi$. The language ATL$^*_{lp}$ is the set of state formulas. The size $|\varphi|$ of a formula φ is the number of distinct subformulas of φ.

The logic ATL$^*_{lp}$ is interpreted over concurrent game trees (CGT) $\mathcal{T} = \langle T, \varepsilon, Lab, \tau \rangle$. For a path formula φ, an initial infinite path π of T (i.e., an ε-play of $\mathcal{T}$) and a position $i \geq 0$, the satisfaction relation $(\mathcal{T}, \pi, i) \models \varphi$, indicating that φ holds at position i along π, is inductively defined as follows (we omit the clauses for the Boolean connectives which are standard):

$$
\begin{array}{ll}
(\mathcal{T}, \pi, i) \models p & \Leftrightarrow \ p \in \mathit{Lab}(\pi(i)), \\
(\mathcal{T}, \pi, i) \models \mathsf{X}\varphi & \Leftrightarrow \ (\mathcal{T}, \pi, i+1) \models \varphi \\
(\mathcal{T}, , \pi, i) \models \mathsf{X}^{-}\varphi & \Leftrightarrow \ i > 0 \text{ and } (\mathcal{T}, \pi, i-1) \models \varphi \\
(\mathcal{T}, \pi, i) \models \varphi_1 \mathsf{U}\, \varphi_2 & \Leftrightarrow \ \text{there is } j \geq i : (\mathcal{T}, \pi, j) \models \varphi_2 \text{ and } (\mathcal{T}, \pi, k) \models \varphi_1 \text{ for all } i \leq k < j \\
(\mathcal{T}, \pi, i) \models \varphi_1 \mathsf{U}^{-}\, \varphi_2 & \Leftrightarrow \ \text{there is } j \leq i : (\mathcal{T}, \pi, j) \models \varphi_2 \text{ and } (\mathcal{T}, \pi, k) \models \varphi_1 \text{ for all } j < k \leq i \\
(\mathcal{T}, \pi, i) \models \langle\langle A \rangle\rangle\varphi & \Leftrightarrow \ \text{for some strategy } f_A \text{ for } A : (\mathcal{T}, \pi', i) \models \varphi \text{ for all } \pi' \in \mathit{out}(\pi[0, i], f_A)
\end{array}
$$

Note that for each node $x \in T$ and ε-play π visiting x, π visits x exactly at position $|x|$ (the distance of node x from the root). For a node x of T, we write $(\mathcal{T}, x) \models \varphi$ to denote that there is an ε-play π visiting x such that $(\mathcal{T}, \pi, |x|) \models \varphi$. Note that if φ is a state formula, then for all ε-plays π and π' which visit node x, it holds that $(\mathcal{T}, \pi, |x|) \models \varphi$ iff $(\mathcal{T}, \pi', |x|) \models \varphi$. A CGT $\mathcal{T}$ satisfies a formula φ (we also say that $\mathcal{T}$ is a model of φ) if $(\mathcal{T}, \varepsilon) \models \varphi$. Two formulas φ and φ' are equivalent if they admit the same models.

It is worth noting that in the valuation of a strategy quantifier $\langle\langle A \rangle\rangle$, while in standard ATL^*, play quantification ranges over the plays consistent with the selected strategy which start at the current node (*local semantics*), in ATL^*_{lp} play quantification ranges over the plays obtained by prefixing the outcomes of the selected strategy from the current node with the history that starts at the root and leads to the current node (*memoryful semantics*). Evidently, for formulas which do not contain past temporal modalities, the standard local ATL^*-semantics is equivalent to the memoryful semantics.[1] Moreover, the known memoryful linear-past extension CTL^*_{lp} [18] of CTL^* corresponds to the fragment of ATL^*_{lp} where only the strategic modalities $\langle\langle Ag \rangle\rangle$ (equivalent to the existential path quantifier E) and $\langle\langle \emptyset \rangle\rangle$ (equivalent to the universal path quantifier A) are exploited. Note that CTL^*_{lp} is interpreted over 2^{AP}-labeled trees which correspond to one-agent CGT. We denote by PATL^* (resp., PCTL^*) the logic having the same syntax as ATL^*_{lp} (resp., CTL^*_{lp}) but interpreted under the local ATL^*-semantics (resp., the local CTL^*-semantics). Intuitively, in PATL^* and PCTL^* the past cannot go beyond the present.

In the following, we also consider the past extension LTL_p of standard (future) LTL [27] which syntactically corresponds to the fragment of ATL^*_{lp} where strategy quantifiers are disallowed. LTL_p formulas ψ over AP are interpreted over infinite words over 2^{AP}. For such words w and positions i, we write $(w, i) \models_{\mathsf{LTL}} \psi$ to denote that ψ holds at position i along w according to the standard LTL (LTL_p) semantics. A *pure past* LTL_p formula is an LTL_p formula which does not contain occurrences of future temporal modalities.

2.3 The logic ATL_{lp}

The logics CTL [10] and ATL [2] are well-known syntactical fragments of CTL^* and ATL^* which have received a lot of attention due to the existence of polynomial-time algorithms which solve the associated (finite) model-checking problem. In particular, CTL (resp., ATL) is obtained from the more expressive logic CTL^* (resp., ATL^*) by requiring that each temporal modality is immediately preceded by a path quantifier (resp., strategy quantifier). In this section, we introduce the ATL-like fragment of ATL^*_{lp} which corresponds to the alternating-time version of the linear-past extension CTL_{lp} of CTL introduced in [18].

Since past is linear, play quantification of past-time modalities is redundant. Thus, in ATL_{lp}, we require the future temporal modalities X, U, and R to be preceded by a strategic

[1] ATL^* formulas φ are usually interpreted over CGS $\mathcal{G}$. However, ATL^* is insensitive to unwinding: $\mathcal{G}$ is a model of φ iff the CGT $\mathit{Unw}(\mathcal{G})$ is a model of φ.

quantifier, but we impose no equivalent restriction on the past temporal modalities. Formally, the set of ATL_{lp} formulas ψ is the fragment of ATL^*_{lp} defined by the following grammar:

$$\psi ::= \top \mid p \mid \neg\psi \mid \psi \vee \psi \mid \mathsf{X}^-\psi \mid \psi\,\mathsf{U}^-\,\psi \mid \langle\langle A\rangle\rangle\mathsf{X}\psi \mid \langle\langle A\rangle\rangle(\psi\mathsf{U}\psi) \mid \langle\langle A\rangle\rangle(\psi\mathsf{R}\psi)$$

Note that for an ATL_{lp} formula ψ and for all ε-plays π and π' which visit a node x of a CGT $\mathcal{T}$, $(\mathcal{T},\pi,|x|) \models \psi$ iff $(\mathcal{T},\pi',|x|) \models \psi$. As an example, let us consider the formula $\psi := \neg\langle\langle A\rangle\rangle\neg\mathsf{G}\langle\langle B\rangle\rangle\mathsf{G}[(\mathit{grant} \rightarrow \mathsf{F}^-\mathit{req}) \wedge \mathsf{G}^-(\mathit{grant} \rightarrow \mathsf{F}^-\mathit{req})]$ which specifies that in each node x reached by a strategy of agents in A, the agents in B can enforce that along a full computation (starting from the root) and visiting x, every grant is preceded by some request.

While ATL^*_{lp} is known to have the same expressiveness as ATL^* [26], ATL_{lp} turns out to be strictly more expressive than ATL. In particular, the following holds.

▶ **Proposition 1** (Expressiveness of ATL_{lp}). *ATL_{lp} is strictly more expressive than ATL and strictly less expressive than ATL^*.*

Proof. We exploit known expressiveness results about the logic CTL_{lp}. In particular, in [18], it is shown that the CTL_{lp} formula $\mathsf{AXAF}(p \wedge \mathsf{X}^-p)$ has no equivalent CTL formula, and the CTL^* formula $\mathsf{EGF}p$ has no equivalent CTL_{lp} formula. Note that over 1-agent CGT (corresponding to labeled trees), a CTL_{lp} formula ψ is equivalent to the ATL_{lp} formula obtained from ψ by replacing the path quantifiers E and A with the strategy quantifiers $\langle\langle Ag\rangle\rangle$ and $\langle\langle\emptyset\rangle\rangle$. It follows that the ATL_{lp} formula $\langle\langle\emptyset\rangle\rangle\mathsf{X}\langle\langle\emptyset\rangle\rangle\mathsf{F}(p \wedge \mathsf{X}^-p)$ cannot expressed in ATL, and the ATL^* formula $\langle\langle Ag\rangle\rangle\mathsf{GF}p$ cannot be expressed in ATL_{lp}. Thus, since ATL_{lp} is a fragment of ATL^*_{lp}, and ATL^*_{lp} and ATL^* have the same expressiveness, the result follows. ◀

We consider the following decision problems:

- *Satisfiability*: has a given ATL_{lp} state formula a model?
- *(Finite) Model Checking*: given a finite CGS $\mathcal{G}$ and an ATL_{lp} state formula ψ, is $\mathit{Unw}(\mathcal{G})$ a model of ψ?

3 Succinctness gap between memoryful past and local past

It is known that PCTL^* and CTL^*_{lp} have the same expressiveness as CTL^* [18] and there are translations from PCTL^* and CTL^*_{lp} into CTL^* of non-elementary complexity [18] based on the separation theorem for LTL_p [12]. On the other hand, the ability to refer to the past makes both PCTL^* and CTL^*_{lp} exponentially more succinct than CTL^*. Analogous results hold for the logics PATL^* and ATL^*_{lp} when compared to ATL^* [26]. Note that as observed in [26], the succinctness gap between the past extensions of CTL^* (resp., ATL^*) and CTL^* (resp., ATL^*) are a consequence of the fact that LTL_p is exponentially more succinct than LTL [22]. Interestingly, and perhaps surprisingly, we can establish a similar result for the logic CTL_{lp} (resp., ATL_{lp}) when compared to the more expressive logic CTL^* (resp., ATL^*).

▶ **Theorem 3.** *CTL_{lp} (resp., ATL_{lp}) can be exponentially more succinct than CTL^* (resp., ATL^*).*

Proof. We focus on ATL_{lp} and ATL^* (a similar result holds for the logics CTL_{lp} and CTL^*). For each $n \geq 1$, let $p_0, p_1, \ldots, p_n$ be $n+1$ atomic propositions, and ψ_n be the LTL_p formula $\mathsf{G}\big(\bigwedge_{i=1}^{i=n}(p_i \leftrightarrow \mathsf{F}^-(\neg\mathsf{X}^-\top \wedge p_i)) \longrightarrow (p_0 \leftrightarrow \mathsf{F}^-(\neg\mathsf{X}^-\top \wedge p_0)\big)$. Note that $|\psi_n| = O(n)$. On the other hand, it is shown in [22] that every LTL formula which is equivalent to ψ_n has size at least $2^{\Omega(n)}$. Now, we observe that $\langle\langle\emptyset\rangle\rangle\psi_n$ is an ATL_{lp} formula. Let φ be an ATL^* formula equivalent to $\langle\langle\emptyset\rangle\rangle\psi_n$, and φ_{LTL} be the LTL formula obtained from φ by removing the occurrences of strategy quantifiers $\langle\langle A\rangle\rangle$. Let Π be the set of CGT over 2^{AP} having exactly one ε-play. Evidently, we have that for each $\mathcal{T} \in \Pi$, $\mathcal{T}$ is a model of $\langle\langle\emptyset\rangle\rangle\psi_n$ (resp., φ) iff $Lab(\pi_{\mathcal{T}})$ is a model of the LTL_p formula ψ_n (resp., LTL formula φ_{LTL}), where $\pi_{\mathcal{T}}$ is the unique ε-play of $\mathcal{T}$. Thus, since φ and $\langle\langle\emptyset\rangle\rangle\psi_n$ are equivalent, and there is a bijection between Π and the set of infinite words over 2^{AP}, we obtain that φ_{LTL} is equivalent to ψ_n, hence, φ_{LTL} has size at least $2^{\Omega(n)}$. This entails that $|\varphi|$ is at least $2^{\Omega(n)}$, and we are done. ◀

To the best of our knowledge, no succinctness gap is known between the 'memoryful' past view in CTL^*_{lp} (resp., ATL^*_{lp}) and the 'local' past view in PCTL^* (resp., PATL^*). In this section, we address this issue by establishing the following.

▶ **Theorem 4.** *CTL^*_{lp} (resp., ATL^*_{lp}) can be exponentially more succinct than PCTL^* (resp., PATL^*).*

For the part of Theorem 4 concerning CTL^*_{lp} and PCTL^*, we prove the following.

▶ **Theorem 5.** *For each $n \geq 1$, there exists a CTL^*_{lp} formula φ_n of size $O(n)$ such that every equivalent PCTL^* formula has size at least $2^{\Omega(n)}$.*

As a corollary of Theorem 5, we deduce the part of Theorem 4 for ATL^*_{lp} and PATL^*.

▶ **Corollary 6.** *Let Ag be a set of agents. For each $n \geq 1$, there exists an ATL^*_{lp} formula φ'_n over Ag of size $O(n)$ such that every equivalent PATL^* formula has size at least $2^{\Omega(n)}$.*

Proof. Fix a set Ag of agents and an agent $ag \in Ag$. Let $n \geq 1$ and φ_n be the CTL^*_{lp} formula of size $O(n)$ satisfying the statement of Theorem 5, and AP_n the set of propositions occurring in φ_n. We denote by φ'_n the ATL^*_{lp} version over Ag of φ_n, i.e., the formula obtained from φ_n, by replacing the path quantifier E (resp., A) with the strategy quantifier $\langle\langle Ag\rangle\rangle$ (resp., $\langle\langle\emptyset\rangle\rangle$). Note that $|\varphi'_n| = |\varphi_n|$. We can associate with each 2^{AP_n}-labeled tree $\mathcal{T} = \langle T, Lab\rangle$ a CGT of the form $f(\mathcal{T}) = \langle T, \varepsilon, Lab, \tau\rangle$ over Ag such that each node in $f(\mathcal{T})$ is controlled by agent ag.

Now, let ψ'_n be a PATL^* formula over Ag and AP_n equivalent to φ'_n. We show that the size of ψ'_n is at least $2^{\Omega(n)}$, hence, the result follows. We denote by ψ_n the PCTL^* formula obtained from ψ'_n by replacing each strategy quantifier $\langle\langle A\rangle\rangle$ with E if $ag \in A$, and with A otherwise. By construction, for each 2^{AP_n}-labeled tree $\mathcal{T} = \langle T, Lab\rangle$, $\mathcal{T}$ is a model of ψ_n (resp., φ_n) iff $f(\mathcal{T})$ is a model of ψ'_n (resp., φ'_n). Thus, since ψ'_n and φ'_n are equivalent, we obtain that the PCTL^* formula ψ_n is equivalent to φ_n. By Theorem 5, the size of ψ_n is at least $2^{\Omega(n)}$, hence, being $|\psi'_n| \geq |\psi_n|$, the result follows. ◀

In the rest of this section, we provide a proof of Theorem 5. Fix $n \geq 1$. Let $AP_n := \{p_0, p_1, \ldots, p_n\}$ be a set consisting of $n+1$ distinct atomic propositions, $AP'_n := AP_n \setminus \{p_0\}$ be the set obtained from AP_n by removing proposition p_0, and $\# \notin AP_n$ be a special atomic proposition. Fix an ordering $a_1, \ldots, a_{2^n}$ of the 2^n distinct symbols of the alphabet $2^{AP'_n}$. An *n-word* is a word of the form $w = b_1 \ldots b_{2^n} \cdot \{\#\}$ over $2^{AP_n \cup \{\#\}}$ of length $2^n + 1$ such that there is a set $K \subseteq [1, 2^n]$ so that for all $j \in [1, 2^n]$, $b_j = a_j$ if $j \notin K$ and $b_j = a_j \cup \{p_0\}$ otherwise. Intuitively an n-word is obtained from $\nu = a_1, \ldots, a_{2^n} \cdot \{\#\}$ by augmenting some non-last symbols in ν with proposition p_0. Note that there are exactly 2^{2^n} distinct n-words. Next, we define the notions of n-block and n-configuration. Let $check \notin AP_n \cup \{\#\}$ be an additional atomic proposition.

▶ **Definition 7** (n-blocks). An n-block is a $2^{AP_n \cup \{check, \#\}}$-labeled tree $\langle T, Lab\rangle$ such that:

- there is a finite path π from the root so that $Lab(\pi)$ is an n-word and each node x of π has exactly one child c_x which is not visited by π. Moreover, there is exactly one infinite path starting from c_x, and this path has label $\{check\}^\omega$.

Intuitively, an n-block is a labeled tree consisting of an n-word augmented with *check*-branches starting from the n-word nodes.

▶ **Definition 8** (n-configurations)**.** An n-configuration is a $2^{AP_n \cup \{check,\#\}}$-labeled tree $\mathcal{T} = \langle T, Lab \rangle$ such that for some $k \geq 1$, there exist $k+1$ n-blocks $\mathcal{T}_0, \ldots, \mathcal{T}_k$ so that $\mathcal{T}$ is obtained by connecting the $k+1$ n-blocks $\mathcal{T}_0, \ldots, \mathcal{T}_k$ as follows:

- the $\{\#\}$-node of $\mathcal{T}_0$ has as children the roots of $\mathcal{T}_1 \ldots \mathcal{T}_k$.

We say that $\mathcal{T}_0$ is the *master* n-block of $\mathcal{T}$, and $\mathcal{T}_1 \ldots, \mathcal{T}_k$ are the *slave* n-blocks of $\mathcal{T}$. The n-configuration $\mathcal{T}$ is *well-formed* if the n-word associated with the master coincides with the n-word of some slave.

We now show that one can construct a CTL^*_{lp} formula φ_n of size $O(n)$ which distinguishes the n-configurations which are well-formed from the non-well-formed ones.

▶ **Lemma 9.** *For each $n \geq 1$, there exists a CTL^*_{lp} formula φ_n over $2^{AP_n \cup \{check,\#\}}$ having size $O(n)$ such that for each n-configuration $\langle T, Lab \rangle$, $\langle T, Lab \rangle$ is a model of φ_n if and only if $\langle T, Lab \rangle$ is well-formed.*

Proof. Given $n \geq 1$, we construct a CTL^*_{lp} formula φ_n which is satisfied by an n-configuration $\mathcal{T}$ iff there exists an infinite path π of $\mathcal{T}$ from the root such that:

(1) π visits the n-word of the master and the n-word of some slave, and

(2) the two n-words visited by π coincide.

The crucial property which allow us to define a CTL^*_{lp} formula of size $O(n)$ satisfying Requirement (2) is that for each node x of π associated with the n-word of the slave, there is a child y of x whose subtree reduces to a path labelled by the word $(\{check\})^\omega$. Thus, the 'memoryful' semantics of the path quantifier E allow us to 'select' the prefix of π until node x, by using the *check*-node y as marker, and to require that such a prefix satisfies the following

- (*) for each node z along the n-word of the master, if the labels of z and x agree on propositions $p_1, \ldots, p_n$, then they also agree on proposition p_0.

Thus, the CTL^*_{lp} formula φ_n is $\mathsf{EF}\big[\# \wedge (\mathsf{XF}\#) \wedge \mathsf{G}\big((\neg\# \wedge \neg check) \rightarrow \mathsf{E}(\mathsf{X}check \wedge \psi_n)\big)\big]$ where ψ_n is an LTL_p formula ensuring Requirement (*) for the selected 2^{AP_n}-labeled node of the selected slave: $\psi_n := \mathsf{F}^-\big[\# \wedge \mathsf{X}^-\mathsf{G}^-\big(\bigwedge_{i=1}^{i=n} \bigwedge_{q_i \in \{p_i, \neg p_i\}} \big(q_i \rightarrow \mathsf{F}(\neg check \wedge \mathsf{X}check \wedge q_i)\big) \longrightarrow \bigwedge_{q_0 \in \{p_0, \neg p_0\}} (q_0 \rightarrow \mathsf{F}(\neg check \wedge \mathsf{X}check \wedge q_0)\big)\big)\big]$. ◀

Next, in order to complete the proof of Theorem 5, we show that for all $n \geq 1$, every PCTL^* formula equivalent to the CTL^*_{lp} formula φ_n of Lemma 9 has size at least $2^{\Omega(n)}$. For this, we exploit the well-known result concerning the translation of CTL^* (and PCTL^* as well) formulas into equivalent parity symmetric alternating tree-automata (parity SATA). SATA [16] are a variation of classical (asymmetric) alternating automata in which it is not necessary to specify the direction (i.e., the choice of the children) of the tree on which a copy is sent. In fact, through existential and universal moves, it is possible to send a copy of the automaton, starting from a node of the input tree, to one or all its successors. We now recall syntax and semantics of parity SATA. For a set X, $\mathbb{B}^+(X)$ denotes the set of *positive* Boolean formulas over X, i.e. Boolean formulas built from elements in X using $\vee$ and $\wedge$ (we also allow the formulas `true` and `false`).

A parity SATA over a finite alphabet Σ is a tuple $\mathcal{A} = \langle \Sigma, q_0, Q, \delta, \Omega \rangle$, where Q is a finite set of states, $q_0 \in Q$ is the initial state, $\delta : Q \times 2^{AP} \rightarrow \mathbb{B}^+(Q \times \{\Box, \Diamond\})$ is the transition function, and $\Omega : Q \mapsto \mathbb{N}$ is a parity acceptance condition over Q assigning to each state an integer (*color*). Intuitively, a target of a move of $\mathcal{A}$ is encoded by an element in $Q \times \{\Box, \Diamond\}$. An atom $(q, \Diamond)$ means that from the current node x of the Σ-labeled input tree, $\mathcal{A}$ moves to some child of x and the state is updated to q. On the other hand, an atom $(q, \Box)$ means that from the current node x, the automaton splits in multiple copies and, for each child x' of x in the input tree, one of such copies moves to node x' and the state is updated to q.

Formally, for a Σ-labeled tree $\mathcal{T} = \langle T, Lab \rangle$, a run of $\mathcal{A}$ over $\mathcal{T}$ is a $(Q \times T)$-labeled $\mathbb{N}$-tree $r = \langle T_r, Lab_r \rangle$, where each node of T_r labelled by (q, x) describes a copy of the automaton that is in state q and reads the node x of T. Moreover, we require that $r(\varepsilon) = (q_0, \varepsilon)$ (initially, the automaton is in state q_0 reading the root node), and for each $y \in T_r$ with $r(y) = (q, x)$, there is a (possibly empty) *minimal* set $H \subseteq Q \times \{\Box, \Diamond\}$ satisfying $\delta(q, Lab(x))$ such that the set $L(y)$ of labels of children of y in T is the smallest set satisfying the following conditions: for all atoms $at \in H$,

- if $at = (q', \Diamond)$, then for some child x' of x in T, $(q', x') \in L(y)$;
- if $at = (q', \Box)$, then for each child x' of x in T, $(q', x') \in L(y)$.

The run r is accepting if for all infinite paths π starting from the root, the smallest color of the states in Q that occur infinitely often along $Lab_r(\pi)$ is *even*. A Σ-labeled tree $\mathcal{T} = \langle T, Lab \rangle$ is accepted by $\mathcal{A}$ if there is an accepting run over $\mathcal{T}$. The following is a well-known result [21].

▶ **Proposition 2** ([21]). *Given a PCTL* formula ψ, one can construct a parity SATA with $2^{O(|\psi|)}$ states accepting the set of models of ψ.*

Theorem 5 directly follows from Lemma 9, Proposition 2, and the following result.

▶ **Lemma 10.** *Let $n \geq 1$ and $\mathcal{A}_n$ be a parity SATA over $2^{AP_n \cup \{check, \#\}}$ accepting the set of models of the CTL$^*_{lp}$ formula φ_n in Lemma 9. Then, $\mathcal{A}_n$ has at least 2^{2^n} states.*

Proof. Let $n \geq 1$ and $\mathcal{A}_n = \langle 2^{AP_n \cup \{check, \#\}}, q_0, Q, \delta, \Omega \rangle$ as in the statement of the lemma. For each state $q \in Q$, we denote by $\mathcal{A}_n^q$ the parity SATA $\langle 2^{AP_n \cup \{check, \#\}}, q, Q, \delta, \Omega \rangle$, i.e., $\mathcal{A}_n^q$ is obtained from $\mathcal{A}_n$ by considering q as initial state instead of q_0. We show the following.

Claim. For each n-word w, there is a state q_w of $\mathcal{A}_n$ such that:
- there is an accepting run of $\mathcal{A}_n^{q_w}$ over the n-block associated with w;
- for each n-word w' distinct from w, there is no accepting run of $\mathcal{A}_n^{q_w}$ over the n-block related to w'.

By the claim above, it follows that there is a bijection between the set of n-words and a subset of the set Q of $\mathcal{A}_n$-states. Thus, since the number of distinct n-words is 2^{2^n}, the result follows. It remains to prove the claim.

An n-configuration $\mathcal{T}$ is *complete* if $\mathcal{T}$ has 2^{2^n} slaves and for each n-word w, there is a slave of $\mathcal{T}$ associated with w. Fix an n-word w and let $\mathcal{T}_w$ be the complete n-configuration whose master matches the n-word w. Since $\mathcal{T}_w$ is well-formed, by hypothesis and Lemma 9, there is an accepting run r_w of $\mathcal{A}_n$ over $\mathcal{T}_w$. Let $x_\#$ be the $\{\#\}$-labeled node of the $\mathcal{T}_w$-master, and $x_1, \ldots, x_{2^{2^n}}$ be the 2^{2^n} children of $x_\#$ which correspond to the roots of the $\mathcal{T}_w$-slaves. Without loss of generality, assume that x_1 is the root of the slave associated with the n-word w. Moreover, let $\mathcal{T}_w^{\setminus w}$ be the n-configuration obtained from $\mathcal{T}_w$ by removing the subtree rooted at node x_1 (i.e., the slave associated with w), and $Q_w^{\Diamond}$ be the set of states associated with the copies of the run r_w reading node x_1 and obtained by *existential* moves. Since $\mathcal{T}_w^{\setminus w}$ is not well-formed, by hypothesis and Lemma 9, there is no accepting run of $\mathcal{A}_n$ over $\mathcal{T}_w^{\setminus w}$.

Since the parity acceptance condition is prefix independent, by construction, for each $q \in Q_w^{\Diamond}$, there exists an accepting run of $\mathcal{A}_n^q$ over the n-block associated with w (in particular, the subtree rooted at the node of the run r_w associated with the copy of $\mathcal{A}_n$ reading node x_1 in state q is such a run). Thus, since w is an arbitrary n-word, in order to prove the claim, it suffices to show that there exists a state $q_w \in Q_w^{\Diamond}$ such that for each n-word w' distinct from w, there is no accepting run of $\mathcal{A}_n^{q_w}$ over the n-block related to w'. We assume the contrary and derive a contradiction. Let $Q_w^{\Diamond} = \{q_1, \ldots, q_k\}$. By hypothesis, for all $i \in [1, k]$, there exists an n-word $w_i \neq w$ and an accepting run r_i of $\mathcal{A}_n^{q_i}$ over the n-block related to w_i. Let r' be the labeled tree obtained from the accepting run r_w by replacing for all $i \in [1, n]$ and node z_i of r_w associated with the copy of $\mathcal{A}_n$ reading node x_1 in state q_i, the subtree rooted at node z_i with a copy of the run r_i. Since for each $i \in [1, k]$, the n-configuration $\mathcal{T}_w^{\setminus w}$ has a slave associated with the n-word w_i, by construction, we obtain that r' is an accepting run of $\mathcal{A}_n$ over $\mathcal{T}_w^{\setminus w}$, which is a contradiction, and the result follows. ◀

4 Decision procedures for ATL$_{lp}$

In this section, we establish that both satisfiability and model-checking for the logic ATL$_{lp}$ are EXPTIME-complete. The upper bounds are obtained by an automata-theoretic framework based on the use of a subclass of Büchi alternating automata for CGS (Büchi ACG) [29], called *ACG with satellites* and introduced in [26]. By translating ATL$_{lp}$ formulas into equivalent Büchi ACG with satellites, we reduce model-checking (resp., satisfiability) of ATL$_{lp}$ to the membership (resp., non-emptiness) problem of such a class of automata. Notice that the symbolic algorithm, based on reachability analysis, used for solving ATL model-checking [2] cannot applied to ATL$_{lp}$. This is because, differently from ATL, the valuation of an ATL$_{lp}$ formula φ at a node x of the unwinding $Unw(\mathcal{G})$ of a finite CGS does not depend only on the state of $\mathcal{G}$ associated with node x, but also depends on the history from the root to node x.

The rest of the section is organized as follows. In Subsection 4.1, we establish a preliminary result concerning the dualization of basic ATL$_{lp}$ formulas which generalizes to linear past a similar result holding for ATL*. In Subsection 4.2, we recall the framework of ACG with satellites, and in Subsection 4.3, we solve satisfiability and model-checking of ATL$_{lp}$ by providing a translation of ATL$_{lp}$ formulas into equivalent Büchi ACG with satellites involving a single exponential blowup.

4.1 Dualization of basic ATL$_{lp}$ formulas

In order to solve the considered decision problems for the logic ATL$_{lp}$, we exploit the following preliminary result (which is known to hold for ATL* [28]).

▶ **Proposition 3.** *Given a basic ATL$_{lp}$ formula $\langle\langle A\rangle\rangle\psi$, a CGT $\mathcal{T}$, and a node x, the following holds, where ν_x is the history from the root to node x: $(\mathcal{T}, x) \models \neg\langle\langle A\rangle\rangle\psi$ if and only if there is a* counter *strategy f_A^c for A such that for all ε-plays $\pi \in out(\nu_x, f_A^c)$, $(\mathcal{T}, \pi, |x|) \models \neg\psi$.*

We now provide a proof of Proposition 3. Given two path formulas φ and φ' of ATL$^*_{lp}$ (note that we consider the more expressive logic ATL$^*_{lp}$), φ and φ' are *congruent* if for every CGT $\mathcal{T}$, ε-play π of $\mathcal{T}$ and position $i \geq 0$, $(\mathcal{T}, \pi, i) \models \varphi$ if and only if $(\mathcal{T}, \pi, i) \models \varphi'$ (note that congruence is a *stronger* requirement than equivalence). We first show that given an ATL$^*_{lp}$ path formula, it is possible to suitably separate past and future temporal modalities.

▶ **Lemma 11.** *Let φ be an ATL$^*_{lp}$ path formula. Then, φ is congruent to a Boolean combination of ATL* formulas and ATL$^*_{lp}$ formulas that correspond to* pure past *LTL$_p$ formulas over the set of propositions $AP \cup \mathcal{H}$, where $\mathcal{H}$ consists of basic ATL* formulas.*

The proof of Lemma 11, which is provided in Appendix A, is based on the well-known separation theorem for LTL_p over infinite words [12], which states that any LTL_p formula can be effectively converted into an equivalent Boolean combination of LTL formulas and pure past LTL_p formulas. We now prove Proposition 3. Let $\langle\langle A\rangle\rangle\psi$, $\mathcal{T}$, x, and ν_x be as in Proposition 3. First, assume that there is a counter strategy f^c_A for A such that for all ε-plays $\pi \in out(\nu_x, f^c_A)$, $(\mathcal{T}, \pi, |x|) \models \neg\psi$. Since for each strategy f_A for A, $out(\nu_x, f^c_A) \cap out(\nu_x, f_A) \neq \emptyset$, we deduce that $(\mathcal{T}, x) \models \neg\langle\langle A\rangle\rangle\psi$.

For the converse direction, assume that $(\mathcal{T}, x) \models \neg\langle\langle A\rangle\rangle\psi$. By Lemma 11, the ATL_{lp} formula ψ is congruent to an ATL^*_{lp} path formula θ of the form

$$\theta := \bigvee_{\ell\in I} (\theta_{p,\ell} \wedge \theta_{f,\ell}) \tag{1}$$

such that $I \neq \emptyset$, for all $\ell \in I$, $\theta_{f,\ell}$ is a path ATL^* formula and $\theta_{p,\ell}$ corresponds to a pure past LTL_p formula over the set of propositions $AP \cup \mathcal{H}$ where $\mathcal{H}$ consists of basic ATL^* formulas. Let J_x be the set of elements $\ell \in I$ such that $(\mathcal{T}, x) \models \theta_{p,\ell}$. If $J_x = \emptyset$, then by Point 1, for each ε-play π visiting node x, we have that $(\mathcal{T}, \pi, |x|) \models \neg\theta$. Thus, since θ is congruent to ψ, the result follows. Now, assume that $J_x \neq \emptyset$. In the proof of Lemma 11 (see Appendix A), we exploit the fact that $\langle\langle A\rangle\rangle \bigvee_{\ell\in I} (\theta_{p,\ell} \wedge \theta_{f,\ell})$ is congruent to $\bigvee_{J\subseteq I | J\neq\emptyset} ((\bigwedge_{\ell\in J} \theta_{p,\ell}) \wedge \langle\langle A\rangle\rangle(\bigvee_{\ell\in J} \theta_{f,\ell}))$. By hypothesis, $(\mathcal{T}, x) \models \neg\langle\langle A\rangle\rangle\psi$. Hence, by definition of J_x, it follows that

$$(\mathcal{T}, x) \models \neg\langle\langle A\rangle\rangle(\textstyle\bigvee_{\ell\in J_x} \theta_{f,\ell})$$

Since $\langle\langle A\rangle\rangle(\bigvee_{\ell\in J_x} \theta_{f,\ell})$ is a basic ATL^* formula, by [28], there exists a counter strategy f^c_A for agents in A such that for all ε-plays $\pi \in out(\nu_x, f^c_A)$ and $\ell \in J_x$, $(\mathcal{T}, \pi, |x|) \models \neg\theta_{f,\ell}$.[2] Moreover, since $\theta_{p,\ell}$ is a pure past LTL_p formula over $AP \cup \mathcal{H}$, where $\mathcal{H}$ consists of basic ATL^* formulas, by definition of J_x, for all $\ell \in I \setminus J_x$ and ε-plays π visiting x, $(\mathcal{T}, \pi, |x|) \models \neg\theta_{p,\ell}$. Thus, by Point 1 and since θ is congruent to ψ, we obtain that for all ε-plays $\pi \in out(\nu_x, f^c_A)$, $(\mathcal{T}, \pi, |x|) \models \neg\psi$, which concludes the proof of Proposition 3.

4.2 Automata for ATL_{lp}

In this section, we recall the class of ACG [29] and the subclass of *ACG with satellites* [26]. ACG generalize the class of symmetric alternating automata (recalled in Section 3) by branching universally or existentially over all successors that result from the agents' decisions. Formally, a Büchi ACG over 2^{AP} and Ag is a tuple $\mathcal{A} = \langle 2^{AP}, q_0, Q, \delta, F\rangle$, where Q is a finite set of states, $q_0 \in Q$ is the initial state, $\delta : Q \times 2^{AP} \to \mathbb{B}^+(Q \times \{\Box, \Diamond\} \times 2^{Ag})$ is the transition function, and $F \subseteq Q$ is a Büchi acceptance condition on Q. The transition function δ maps a state and an input letter to a positive Boolean combination of universal atoms $(q, \Box, A)$ which refer to *all* successors states for some A-decision, and existential atoms $(q, \Diamond, A)$ which refer to *some* successor state for all A-decisions.

We interpret the Büchi ACG $\mathcal{A}$ over CGT $\mathcal{T} = \langle T, \varepsilon, Lab, \tau\rangle$ on AP and Ag. A run of $\mathcal{A}$ over $\mathcal{T}$ is a $(Q \times T)$-labeled $\mathbb{N}$-tree $r = \langle T_r, Lab_r\rangle$, where each node of T_r labelled by (q, x) describes a copy of the automaton that is in the state q and reads the node x of T. Moreover, we require that $Lab_r(\varepsilon) = (q_0, \varepsilon)$ (initially, the automaton is in state q_0 reading the root node), and for each $y \in T_r$ with $Lab_r(y) = (q, x)$, there is a set $H \subseteq Q \times \{\Box, \Diamond\} \times 2^{Ag}$ such

[2] The result in [28] is based on the well-known determinacy of two-players turn-based games with LTL objectives.

that H is model of $\delta(q, Lab(x))$ and the set L of labels associated with the children of y in T_r minimally satisfies the following conditions:

- for all universal atoms $(q', \Box, A) \in H$, there is some A-decision d_A such that for all the children x' of x in $\mathcal{T}$ which are consistent with d_A, $(q', x') \in L$;
- for all existential atoms $(q', \Diamond, A) \in H$ and for all A-decisions d_A, there is some child x' of x in $\mathcal{T}$ which is consistent with d_A such that $(q', x') \in L$.

The run r is accepting if for all infinite paths π starting from the root, $Lab_r(\pi)$ visits infinitely often elements in F. The language $\mathcal{L}(\mathcal{A})$ accepted by $\mathcal{A}$ consists of the CGT $\mathcal{T}$ over AP and Ag such that there is an accepting run of $\mathcal{A}$ over $\mathcal{T}$.

In the following, we also consider standard nondeterministic and deterministic word automata (NWA and DWA) with *no* acceptance condition (safety NWA and safety DWA) running on words over a finite alphabet Σ. Recall that a safety NWA is a tuple $\mathcal{A} = \langle \Sigma, q_0, Q, \delta \rangle$, where q_0 and Q are as for ACG, and $\delta : Q \times \Sigma \mapsto 2^Q$ is the transition function. The automaton is deterministic if for each state q and input symbol σ, $\delta(q, \sigma)$ is a singleton. A run π over an input $w \in \Sigma^\omega$ is an infinite sequence of states $\pi = q_0, q_1, \ldots$ (starting from the initial state) such that $q_{i+1} \in \delta(q_i, w(i))$ for all $i \geq 0$.

In the translation of ATL_{lp} formulas into equivalent ACG, the main technical obstacle is the handling of (memoryful) linear past which enables a reference to histories since the root of a CGT. This requires an automaton to remember the past. A simple solution would consist of using a two-way extension of ACG, but for such a class of automata, the membership and the non-emptiness problem would result harder than the ones for (one-way) ACG. Instead, we adopt the approach for CTL^*_{lp} [18] and ATL^*_{lp} [26] based on the use of alternating automata augmented with *satellites*. Specifically, ACG with satellites [26] represent a subclass of ACG in which the state space can be partitioned into two components, one of which (the *satellite*) is independent of the other, has no influence on the acceptance, and runs on all the branches of the input CGT by maintaining information about the past.

Formally, a Büchi ACG equipped with a satellite is a pair $(\mathcal{A}, \mathcal{U})$, where $\mathcal{U} = \langle 2^{AP}, s_0, S, \delta_S \rangle$, the satellite, is a safety NWA, while $\mathcal{A}$ (the main automaton) is a Büchi ACG $\mathcal{A} = \langle 2^{AP}, q_0, Q, \delta, F \rangle$ whose transition function is of the form $\delta : Q \times 2^{AP} \times S \to \mathbb{B}^+(Q \times \{\Box, \Diamond\} \times 2^{Ag})$. Intuitively, when a copy of $\mathcal{A}$ reads a node x of the input with label σ, a possible move is performed in two phases. In the first phase, the associated copy of $\mathcal{U}$ reads the letter σ and updates its state. In the second phase, $\mathcal{A}$ reads σ and the updated state of $\mathcal{U}$, and updates its state. Formally, $(\mathcal{A}, \mathcal{U})$ is equivalent to the ordinary Büchi ACG $\mathcal{A}' = \langle 2^{AP}, (q_0, s_0), Q \times S, \delta', F \times S \rangle$, where for all $(q, s) \in Q \times S$ and $\sigma \in 2^{AP}$, $\delta'((q, s), \sigma) = \bigvee_{s' \in \delta_S(s,\sigma)} f(q, \sigma, s')$ with $f(q, \sigma, s')$ being obtained from $\delta(q, \sigma, s')$ by replacing each atom $(q', \Box, A)$ (resp., $(q', \Diamond, A)$) with $((q', s'), \Box, A)$ (resp., $((q', s'), \Diamond, A)$).

The separation of the satellite $\mathcal{U}$ from the main automaton $\mathcal{A}$ allows a tighter analysis of the complexity of the nonemptiness problem. If $\mathcal{U}$ is *deterministic*, then the resulting exponential blow-up in the alternation removal (used for checking non-emptiness) only concerns the states of the main automaton. In particular, by [18, 26], the following holds.

▶ **Proposition 4** ([18, 26]). *The non-emptiness problem of a Büchi ACG $\mathcal{A}$ with n states and a* deterministic *satellite with n_S states can be solved in time $2^{O(n \log n_S + n^2 \log n)}$.*

In Subsection 4.3, we provide a translation of ATL_{lp} formulas φ into equivalent Büchi ACG accepting the set of models of φ. In such a way, model-checking against ATL_{lp} is reduced to checking for a finite CGS $\mathcal{G}$ and a Büchi ACG $\mathcal{A}$, whether $Unw(\mathcal{G})$ is accepted by $\mathcal{A}$. By standard arguments (see e.g. [21]), checking whether $Unw(\mathcal{G})$ is accepted by $\mathcal{A}$ can be reduced in polynomial-time to check non-emptiness of a Büchi alternating word automaton over a

1-letter input alphabet having $n \cdot m$ states, where n (resp., m) is the number of states in $\mathcal{G}$ (resp., $\mathcal{A}$). By [19], the latter problem can be solved in time $O((n \cdot m)^2)$.

▶ **Proposition 5.** *Given a Büchi ACG $\mathcal{A}$ with m states and a finite CGS $\mathcal{G}$ with n states, checking whether $Unw(\mathcal{G})$ is accepted by $\mathcal{A}$ can be done in time $O((n \cdot m)^2)$.*

4.3 From ATL_{lp} to Büchi ACG with satellites

In this section, in order to solve ATL_{lp} satisfiability, we show how to translate a given ATL_{lp} formula Φ into a Büchi ACG $\mathcal{A}_\Phi$ with $O(|\Phi|)$ main states equipped with a *deterministic* satellite $\mathcal{U}_\Phi$ having $2^{O(|\Phi|)}$ states accepting extended versions of the models of Φ (*extended basic models*). For the model-checking problem, we convert the pair $(\mathcal{A}_\Phi, \mathcal{U}_\Phi)$ into a ACG $\mathcal{A}'_\Phi$ with $O(|\Phi|)$ main states equipped with a nondeterministic satellite $\mathcal{U}'_\Phi$ having $2^{O(|\Phi|)}$ states accepting the set of models of Φ. We now proceed with the technical details.

Fix an ATL_{lp} *state* formula Φ over AP, and let B_Φ be the set of basic subformulas of Φ, and $BFL(\Phi)$ be the set of *first-level basic subformulas* of Φ, i.e. the basic subformulas of Φ for which there is an occurrence in Φ which is not in the scope of strategy quantifiers. Note that an ATL_{lp} formula ψ can be seen as a *pure past* LTL_p formula, denoted $[\psi]_{\mathsf{LTL}_p}$, over the set AP augmented with the set $BFL(\psi) \subseteq B_\psi$ of first-level basic subformulas of ψ. In particular, if ψ is a state formula, then $[\psi]_{\mathsf{LTL}_p}$ is a propositional formula over $AP \cup BFL(\psi)$.

For a finite set B disjoint from AP and a CGT $\mathcal{T} = \langle T, \varepsilon, Lab, \tau\rangle$ over AP, a B-labeling extension of $\mathcal{T}$ is a CGT over $AP \cup B$ of the form $\langle T, \varepsilon, Lab', \tau\rangle$, where $Lab'(x) \cap AP = Lab(x)$ for all $x \in T$. A CGT $\mathcal{T}_\Phi$ over $AP \cup B_\Phi$ is called an *extended basic model of* Φ iff the following holds, where Lab is the labeling of $\mathcal{T}_\Phi$:

- for each $\langle\langle A\rangle\rangle\psi \in B_\Phi$ and node x of $\mathcal{T}_\Phi$, $(\mathcal{T}_\Phi, x) \models \langle\langle A\rangle\rangle\psi$ if and only if $\langle\langle A\rangle\rangle\psi \in Lab(x)$;
- $Lab(\varepsilon)$ is a model of the propositional formula $[\Phi]_{\mathsf{LTL}_p}$.

Evidently, by the semantics of ATL_{lp}, the following holds.

▶ Remark. Let $\mathcal{T}$ be a CGT over AP and Φ be an ATL_{lp} state formula over AP. Then:

- $\mathcal{T}$ is a model of Φ *iff* there exists a B_Φ-labeling extension of $\mathcal{T}$ which is an extended basic model of Φ;
- there is at most one B_Φ-labeling extension of $\mathcal{T}$ which is an extended basic model of Φ.

Next, we provide a characterization of the set of extended basic models of the given state formula Φ based on a set of conditions which can be easily checked by Büchi ACG equipped with deterministic satellites. Let ψ be a pure past LTL_p formula over the set of propositions $AP \cup B_\Phi$. The closure $cl(\psi)$ of ψ is the set of pure past LTL_p formulas consisting of the subformulas of ψ and their negations (we identify $\neg\neg\psi$ with ψ). Note that $\psi, \neg\psi \in cl(\psi)$ and $|cl(\psi)| = O(|\psi|)$. It is worth noting that in the previous definition, elements in B_Φ are considered as atomic propositions. By an adaptation of the well-known translation of LTL formulas into equivalent generalized Büchi NWA [31], we obtain the following result (for details, see Appendix B).

▶ **Proposition 6.** *Let ψ be a pure past LTL_p formula over $AP \cup B_\Phi$. Then, one can construct a safety DWA $\mathcal{D}_\psi = \langle 2^{AP \cup B_\Phi}, q_0, Q \cup \{q_0\}, \delta\rangle$ such that $Q \subseteq 2^{cl(\psi)}$, there is no transition leading to q_0, and the following holds for each* infinite *word w over $2^{AP \cup B_\Phi}$:*

- *let $q_0\, C_0\, C_1 \ldots$ be the unique run of $\mathcal{D}_\psi$ over w. Then, for all $i \geq 0$ and $\theta \in cl(\psi)$, $\theta \in C_i$ if and only if $(w, i) \models \theta$.*

We are now ready to provide a characterization of the extended basic models of the ATL_{lp} state formula Φ. For each ATL_{lp} subformula ψ of Φ, let $\mathcal{D}_\psi$ be the safety DWA over $2^{AP \cup B_\Phi}$

of Proposition 6 for the pure past LTL_p formula $[\psi]_{\mathsf{LTL}_p}$. For each finite word w over $2^{AP\cup B_\Phi}$, we denote by $\mathcal{D}_\psi(w)$ the state reached by $\mathcal{D}_\psi$ on reading w.

▶ **Definition 12** (Well-formedness)**.** Let $\mathcal{T}$ be a CGT over $AP \cup B_\Phi$ with labeling Lab and $\langle\langle A\rangle\rangle\psi \in B_\Phi$ be a basic subformula of Φ. For each node x of $\mathcal{T}$, let ν_x be the track of $\mathcal{T}$ from the root to node x.

- $\mathcal{T}$ is *positively well-formed with respect to* $\langle\langle A\rangle\rangle\psi$ if for all nodes x such that $\langle\langle A\rangle\rangle\psi \in Lab(x)$, there is a strategy f_A for A such that for all the outcomes $\pi \in out(\nu_x, f_A)$:
  - Case $\psi = \mathsf{X}\psi_1$: $\psi_1 \in \mathcal{D}_{\psi_1}(Lab(\pi[0, |x|+1]))$.
  - Case $\psi = \psi_1\mathsf{U}\psi_2$: there is $j \geq |x|$ such that $\psi_2 \in \mathcal{D}_{\psi_2}(Lab(\pi[0,j]))$ and $\psi_1 \in \mathcal{D}_{\psi_1}(Lab(\pi[0,k]))$ for all $k \in [|x|, j-1]$.
  - Case $\psi = \psi_1\mathsf{R}\psi_2$: for all $j \geq |x|$ either $\psi_2 \in \mathcal{D}_{\psi_2}(Lab(\pi[0,j]))$ or $\psi_1 \in \mathcal{D}_{\psi_1}(Lab(\pi[0,k]))$ for some $k \in [|x|, j-1]$.
- $\mathcal{T}$ is *negatively well-formed with respect to* $\langle\langle A\rangle\rangle\psi$ if for all nodes x such that $\langle\langle A\rangle\rangle\psi \notin Lab(x)$, there is a counter strategy f^c_A for A such that for all the outcomes $\pi \in out(\nu_x, f^c_A)$:
  - Case $\psi = \mathsf{X}\psi_1$: $\neg\psi_1 \in \mathcal{D}_{\psi_1}(Lab(\pi[0, |x|+1]))$.
  - Case $\psi = \psi_1\mathsf{U}\psi_2$: for all $j \geq |x|$ either $\neg\psi_2 \in \mathcal{D}_{\psi_2}(Lab(\pi[0,j]))$ or $\neg\psi_1 \in \mathcal{D}_{\psi_1}(Lab(\pi[0,k]))$ for some $k \in [|x|, j-1]$.
  - Case $\psi = \psi_1\mathsf{R}\psi_2$: there is $j \geq |x|$ such that $\neg\psi_2 \in \mathcal{D}_{\psi_2}(Lab(\pi[0,j]))$ and $\neg\psi_1 \in \mathcal{D}_{\psi_1}(Lab(\pi[0,k]))$ for all $k \in [|x|, j-1]$.

▶ **Lemma 13.** *[Characterization of extended basic models of Φ] Let $\mathcal{T}$ be a CGT with labeling Lab over $AP \cup B_\Phi$ such that $Lab(\varepsilon)$ is a model of the propositional formula $[\Phi]_{\mathsf{LTL}_p}$. Then, $\mathcal{T}$ is an extended basic model of Φ* if and only if *for each basic subformula $\langle\langle A\rangle\rangle\psi \in B_\Phi$, $\mathcal{T}$ is both positively and negatively well-formed with respect to $\langle\langle A\rangle\rangle\psi$.*

Lemma 13 easily follows from the dualization result (Proposition 3) and Proposition 6 (see Appendix C). Based on the characterization Lemma, we obtain the following result.

▶ **Theorem 14.** *Given an ATL_{lp} state formula Φ, one can build in single exponential time a Büchi ACG $\mathcal{A}_\Phi$, equipped with a deterministic satellite $\mathcal{U}_\Phi$, that accepts the set of extended basic models of Φ. Moreover, $\mathcal{A}_\Phi$ has $O(|\Phi|)$ states and $\mathcal{U}_\Phi$ has $2^{O(|\Phi|)}$ states.*

Proof. Let $\mathcal{F}$ be the set of subformulas ψ of Φ such that there exists a basic subformula of Φ having one of the following forms: $\langle\langle A\rangle\rangle\mathsf{X}\psi$ or $\langle\langle A\rangle\rangle(\theta\mathcal{O}\psi)$ or $\langle\langle A\rangle\rangle(\psi\mathcal{O}\theta)$ where $\mathcal{O} \in \{\mathsf{U}, \mathsf{R}\}$. Then, the deterministic satellite $\mathcal{U}_\Phi$ is the synchronous product of the safety DWA of Proposition 6 associated with the formulas $[\psi]_{\mathsf{LTL}_p}$ where $\psi \in \mathcal{F}$. For a state s of $\mathcal{U}_\Phi$ and $\psi \in \mathcal{F}$, let $s[\psi]$ be the component of the state s associated with the automaton $\mathcal{D}_\psi$. If s is not initial (hence, $s[\psi] \subseteq cl([\psi]_{\mathsf{LTL}_p})$), we write $P_+(s,\psi)$ to denote `true` if $\psi \in s[\psi]$, and `false` otherwise. Dually, we write $P_-(s,\psi)$ to denote `true` if $\neg\psi \in s[\psi]$, and `false` otherwise.

Next, we define the Büchi ACG $\mathcal{A}_\Phi$. We first build for each basic subformula b of Φ, two Büchi ACG $\mathcal{A}_b$ and $\mathcal{A}_{\neg b}$ such that $\mathcal{A}_b$ (resp., $\mathcal{A}_{\neg b}$), equipped with the satellite $\mathcal{U}_\Phi$, accepts the set of CGT over $AP \cup B_\Phi$ which are positively (resp., negatively) well-formed with respect to b. Then $\mathcal{A}_\Phi$ is obtained by 'composing' the automata $\mathcal{A}_b$ and $\mathcal{A}_{\neg b}$ for all $b \in B_\Phi$.

Fix $b \in B_\Phi$. Then, $\mathcal{A}_b = \langle 2^{AP\cup B_\Phi}, q_b, \{q_b, b\}, \delta_b, F_b\rangle$ consists of two states b and q_b. State q_b is always accepting, i.e. $q_b \in F_b$, while state b is in F_b iff b is of the form $\langle\langle A\rangle\rangle(\psi_1\mathsf{R}\psi_2)$. The transition function δ_b is defined as follows, where s is a state of $\mathcal{U}_\Phi$:

$$\delta_b(q_b,\sigma,s) \quad = \begin{cases} (q_b,\Box,\emptyset) & \text{if } b \notin \sigma \\ (q_b,\Box,\emptyset) \wedge (b,\Box,A) & \text{if } b \in \sigma \text{ and } b \text{ is of the form } \langle\langle A\rangle\rangle\mathsf{X}\psi_1 \\ (q_b,\Box,\emptyset) \,\wedge\, \delta_b(b,\sigma,s) & \text{otherwise} \end{cases}$$

$$\delta_b(b,\sigma,s) \quad = \begin{cases} P_+(s,\psi_1) & \text{if } b = \langle\langle A\rangle\rangle \mathsf{X}\psi_1 \\ P_+(s,\psi_2) \vee (P_+(s,\psi_1) \wedge (b,\Box,A)) & \text{if } b = \langle\langle A\rangle\rangle(\psi_1 \mathsf{U} \psi_2) \\ P_+(s,\psi_2) \wedge (P_+(s,\psi_1) \vee (b,\Box,A)) & \text{if } b = \langle\langle A\rangle\rangle(\psi_1 \mathsf{R} \psi_2) \end{cases}$$

Intuitively, the automaton $\mathcal{A}_b$ uses the part $(q_b, \Box, \emptyset)$ of the transition function to traverse every node in an input CGT $\mathcal{T}$. Additionally, whenever the basic subformula $b = \langle\langle A\rangle\rangle\psi$ is in the label of the current input node x and $\mathcal{A}_b$ is in its initial state q_b, $\mathcal{A}_b$ guesses a strategy f_A of $\mathcal{T}$ for the coalition A and check that for all the plays $\pi \in out(\nu_x, f_A)$, $Lab(\pi)$ satisfies the conditions of Definition 12 (which depend on the form of ψ). This check is done by using the part $(b, \Box, A)$ of the transition function which allows to send copies of $\mathcal{A}_b$, all of them in state b, to all and only the children of the current node which are consistent with the guessed strategy f_A, and by consulting the current state s of the satellite (i.e., the state reached by $\mathcal{U}_\Phi$ on reading the labeling of the track from the root to the current node). The construction of the ACG $\mathcal{A}_{\neg b} = \langle 2^{AP \cup B_\Phi}, q_{\neg b}, \{q_{\neg b}, \neg b\}, \delta_{\neg b}, F_{\neg b}\rangle$ is similar but we now use $P_-(s,\psi_i)$ instead of $P_+(s,\psi_i)$, and we use atoms of the form $(q, \Diamond, A)$ for selecting plays of counter strategies for A in the input CGT. The ACG $\mathcal{A}_\Phi$ is then defined as follows, where we assume that the various automata $\mathcal{A}_b$ and $\mathcal{A}_{\neg b}$ with $b \in B_\Phi$ have no state in common: $\mathcal{A}_\phi = \langle 2^{AP \cup B_\Phi}, q_0, Q \cup q_0, \delta, F\rangle$, where Q (resp., F) consists of the states (resp., accepting states) of $\mathcal{A}_b$ and $\mathcal{A}_{\neg b}$ for all $b \in B_\Phi$, q_0 is a fresh initial state, and the transition function for the states $q \neq q_0$ is inherited from the respective ACG. For the initial state q_0, $\delta(q_0,\sigma,s)$, where s is a satellite state, is defined as follows: $\delta(q_0,\sigma,s) = \texttt{false}$ if σ is not a model of the propositional formula $[\Phi]_{\mathsf{LTL}}$, and $\delta(q_0,\sigma,s) = \bigwedge_{b \in B_\Phi \cap \sigma} \delta_b(q_b,\sigma,s) \wedge \bigwedge_{b \in B_\Phi \setminus \sigma} \delta_{\neg b}(q_{\neg b},\sigma,s)$ otherwise. By construction and Lemma 13, for every CGT $\mathcal{T}$ over $AP \cup B_\Phi$, $\mathcal{A}_\Phi$ accepts $\mathcal{T}$ iff $\mathcal{T}$ is a basic extended model of Φ. Moreover, by Proposition 6, $\mathcal{A}_\Phi$ has $O(|\Phi|)$ states and $\mathcal{U}_\Phi$ has $2^{O(|\Phi|)}$ states, which concludes the proof of Theorem 14. ◀

For solving satisfiability for Φ, we can check the pair $(\mathcal{A}_\Phi, \mathcal{U}_\Phi)$ of Theorem 14 for non-emptiness. However, for solving the model-checking problem against ATL_{lp}, we need automata running on CGT over 2^{AP}. By slightly adapting the construction of Theorem 14, we obtain the following result (for a proof, see Appendix D).

▶ **Theorem 15.** *Given an ATL_{lp} state formula Φ, one can build in single exponential time a Büchi ACG $\mathcal{A}'_\Phi$ over 2^{AP}, equipped with a* nondeterministic *satellite $\mathcal{U}'_\Phi$, that accepts the set of models of Φ. Moreover, $\mathcal{A}'_\Phi$ has $O(|\Phi|)$ states and $\mathcal{U}'_\Phi$ has $2^{O(|\Phi|)}$ states.*

We can show that model-checking against ATL_{lp} is EXPTIME-hard by a polynomial-time reduction from the acceptance problem for linearly-bounded alternating Turing Machines (see Appendix E). Since ATL_{lp} subsumes ATL and satisfiability of ATL is EXPTIME-complete [2], by Propositions 4-5 and Theorems 14-15, we obtain the main result of this section.

▶ **Corollary 16.** *Satisfiability and model-checking against ATL_{lp} are both* EXPTIME*-complete.*

5 Conclusions

While it is well-known that linear past exponentially increases the succinctness of temporal logic formulas, very few is known about the succinctness gap between different linear-past semantics. In this paper, we have partially addressed this issue. Moreover, we have investigated a memoryful linear-past extension of ATL, which expressively strictly lies between ATL and ATL*, and for which satisfiability and model-checking problems are exponentially less expansive than the ones for ATL*. As future work, we aim to investigate memoryful linear-past extensions of meaningful and elementary fragments (such as One-Goal Strategy Logic) of Strategy Logic [9, 25], a well-known powerful framework for reasoning explicitly about strategic behaviors.

References

1 T. Ågotnes, V. Goranko, and W. Jamroga. Alternating-time temporal logics with irrevocable strategies. In *TARK'07*, pages 15–24, 2007.

2 R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.

3 C. Baier and J.P. Katoen. *Principles of Model Checking*. The MIT Press, 2008.

4 L. Bozzelli. The complexity of CTL* + linear past. In *FOSSACS'08*, LNCS 4962, pages 186–200. Springer, 2008.

5 L. Bozzelli and A. Murano. On the complexity of ATL and ATL* module checking. In *GandALF'17*, EPTCS 256, pages 268–282, 2017.

6 T. Brihaye, A. Da Costa Lopes, F. Laroussinie, and N. Markey. ATL with strategy contexts and bounded memory. In *LFCS'09*, LNCS 5407, pages 92–106. Springer, 2009.

7 N. Bulling, W. Jamroga, and M. Popovici. Agents with truly perfect recall in alternating-time temporal logic. In *AAMAS'14*, pages 1561–1562. IFAAMAS/ACM, 2014.

8 A.K. Chandra, D.C. Kozen, and L.J. Stockmeyer. Alternation. *Journal of the ACM*, 28(1):114–133, 1981.

9 K. Chatterjee, T. A. Henzinger, and N. Piterman. Strategy logic. *Inf. Comput.*, 208(6):677–693, 2010.

10 E.M. Clarke and E.A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Proceedings of Workshop on Logic of Programs*, LNCS 131, pages 52–71. Springer-Verlag, 1981.

11 E.A. Emerson and J.Y. Halpern. "Sometimes" and "Not Never" revisited: on branching versus linear time temporal logic. *Journal of the ACM*, 33(1):151–178, 1986.

12 D. M. Gabbay. The declarative past and imperative future: Executable temporal logic for interactive systems. In *Temporal Logic in Specification*, LNCS 398, pages 409–448. Springer, 1987.

13 D.P. Guelev, C. Dima, and C. Enea. An alternating-time temporal logic with knowledge, perfect recall and past: axiomatisation and model-checking. *Journal of Applied Non-Classical Logics*, 21(1):93–131, 2011.

14 Th. Hafer and W. Thomas. Computation tree logic CTL* and path quantifiers in the monadic theory of the binary tree. In *Proc. 14th ICALP*, LNCS 267, pages 269–279. Springer, 1987.

15 W. Jamroga and A. Murano. Module checking of strategic ability. In *AAMAS'15*, pages 227–235. ACM, 2015.

16 D. Janin and I. Walukiewicz. Automata for the modal mu-calculus and related results. In *MFCS'95*, LNCS 969, pages 552–562. Springer, 1995.

17 O. Kupferman and A. Pnueli. Once and For All. In *LICS'95*, pages 25–35. IEEE Comp. Soc. Press, 1995.

18 O. Kupferman, A. Pnueli, and M. Y. Vardi. Once and for all. *J. Comput. Syst. Sci.*, 78(3):981–996, 2012.

19 O. Kupferman and M. Y. Vardi. Weak alternating automata and tree automata emptiness. In *STOC'98*, pages 224–233. ACM, 1998.

20 O. Kupferman and M.Y. Vardi. Module checking. In *CAV'96*, LNCS 1102, pages 75–86. Springer, 1996.

21 O. Kupferman, M.Y. Vardi, and P. Wolper. An Automata-Theoretic Approach to Branching-Time Model Checking. *Journal of the ACM*, 47(2):312–360, 2000.
22 F. Laroussinie, N. Markey, and P. Schnoebelen. Temporal logic with forgettable past. In *LICS'02*, pages 383–392. IEEE Computer Society, 2002.
23 F. Laroussinie and P. Schnoebelen. A hierarchy of temporal logics with past. *Theor. Comput. Sci.*, 148(2):303–324, 1995.
24 F. Laroussinie and P. Schnoebelen. Specification in CTL+past for verification in CTL. *Inf. Comput.*, 156(1-2):236–263, 2000.
25 F. Mogavero, A. Murano, G. Perelli, and M. Y. Vardi. Reasoning about strategies: On the model-checking problem. *ACM Trans. Comput. Log.*, 15(4):34:1–34:47, 2014.
26 F. Mogavero, A. Murano, and M.Y. Vardi. Relentful strategic reasoning in alternating-time temporal logic. *J. Log. Comput.*, 26(5):1663–1695, 2016.
27 A. Pnueli. The temporal logic of programs. In *FOCS'77*, pages 46–57. IEEE, 1977.
28 S. Schewe. ATL* Satisfiability is 2EXPTIME-complete. In *ICALP'08*, LNCS 5126, pages 373–385. Springer, 2008.
29 S. Schewe and B. Finkbeiner. Satisfiability and finite model property for the alternating-time *mu*-calculus. In *CSL'06*, LNCS 4207, pages 591–605. Springer, 2006.
30 R. S. Streett. Propositional dynamic logic of looping and converse. In *STOC'81*, pages 375–383. ACM, 1981.
31 M. Y. Vardi and P. Wolper. Reasoning about infinite computations. *Inf. Comput.*, 115(1):1–37, 1994.
32 M.Y. Vardi. A temporal fixpoint calculus. In *POPL'88*, pages 250–259. ACM, 1988.
33 M.Y. Vardi. Reasoning about the past with two-way automata. In *ICALP'98*, LNCS 1443, pages 628–641. Springer, 1998.

Appendix

A Proof of Lemma 11

For an ATL^*_{lp} path formula φ, let $BFL(\varphi)$ be the set of *first-level basic subformulas* of φ, i.e. the basic subformulas of φ for which there is an occurrence in φ which is not in the scope of strategy quantifiers. We first show that, under the assumption that the first-level basic subformulas are in ATL^*, it is possible to separate past and future temporal modalities.

▶ **Lemma 17.** *Let $\langle\!\langle A\rangle\!\rangle\varphi$ be a basic ATL^*_{lp} formula s.t. $BFL(\varphi)$ consists of ATL^* formulas. Then, $\langle\!\langle A\rangle\!\rangle\varphi$ and φ are congruent to Boolean combinations of ATL^* formulas and ATL^*_{lp} formulas that correspond to* pure past LTL_p *formulas over the set of propositions $AP \cup BFL(\varphi)$.*

Proof. Let $\overline{AP} = AP \cup BFL(\varphi)$. By hypothesis, $BFL(\varphi)$ is a set of ATL^* formulas.

Given a CGT $\mathcal{T}$ over AP with propositional labeling Lab and an ε-play π of $\mathcal{T}$, we denote by $\pi_{\overline{AP}}$ the infinite word over $2^{\overline{AP}}$ defined as follows for every position $i \geq 0$:

- $\pi_{\overline{AP}}(i) \cap AP = Lab(\pi(i))$;
- $\pi_{\overline{AP}}(i) \cap BFL(\varphi) = \{\psi \in BFL(\varphi) \mid (\mathcal{T}, \pi, i) \models \psi\}$.

Now, the path formula φ over AP can be seen as an LTL_p formula over $\overline{AP}$. By definitions of the infinite words $\pi_{\overline{AP}}$ and the LTL_p and ATL^*_{lp} semantics, one can easily show by structural induction that for all CGT $\mathcal{T}$ over AP, ε-plays π of $\mathcal{T}$, and positions $i \geq 0$:

$$(\mathcal{T}, \pi, i) \models \varphi \text{ if and only if } (\pi_{\overline{AP}}, i) \models_{\mathsf{LTL}} \varphi \tag{2}$$

By applying the separation theorem for LTL_p [12], starting from the LTL_p formula φ, one can construct an LTL_p formula θ over $\overline{AP}$ of the form

$$\theta := \bigvee_{\ell \in I} (\theta_{p,\ell} \wedge \theta_{f,\ell}) \tag{3}$$

such that $I \neq \emptyset$, for all $\ell \in I$, $\theta_{p,\ell}$ is a pure past LTL_p formula, $\theta_{f,\ell}$ is an LTL formula, and for all infinite words w over $2^{\overline{AP}}$ and $i \geq 0$, it holds that $(w, i) \models_{\mathsf{LTL}} \varphi$ iff $(w, i) \models_{\mathsf{LTL}} \theta$.

Since θ corresponds to an ATL^*_{lp} formula over AP, by replacing formula φ with θ in Point 2, we obtain that φ and θ are congruent and the given basic ATL^*_{lp} formula $\langle\!\langle A\rangle\!\rangle\varphi$ over AP is congruent to the basic ATL^*_{lp} formula $\langle\!\langle A\rangle\!\rangle \bigvee_{\ell \in I} (\theta_{p,\ell} \wedge \theta_{f,\ell})$. Thus, the statement of Lemma 11 directly follows from the following claim.

Claim. $\psi := \langle\!\langle A\rangle\!\rangle \bigvee_{\ell \in I} (\theta_{p,\ell} \wedge \theta_{f,\ell})$ is congruent to $\psi_B := \bigvee_{J \subseteq I \mid J \neq \emptyset} \big((\bigwedge_{\ell \in J} \theta_{p,\ell}) \wedge \langle\!\langle A\rangle\!\rangle(\bigvee_{\ell \in J} \theta_{f,\ell})\big)$.

Proof of the Claim. Fix a CGT $\mathcal{T}$ over AP, an ε-play π of $\mathcal{T}$, and a position $i \geq 0$.

First, assume that $(\mathcal{T}, \pi, i) \models \psi$. Hence, there is a strategy f_A for A such that the following holds, where Π is the non-empty set of ε-plays $\pi' \in out(\pi[0, i], f_A)$: for all $\pi' \in \Pi$, $(\mathcal{T}, \pi', i) \models \bigvee_{\ell \in I} (\theta_{p,\ell} \wedge \theta_{f,\ell})$. Let J be the non-empty subset of I consisting of the elements $\ell \in I$ such that $(\mathcal{T}, \pi', i) \models \theta_{p,\ell} \wedge \theta_{f,\ell}$ for some $\pi' \in \Pi$. Since $\theta_{p,\ell}$ corresponds to a pure past LTL_p formula over $\overline{AP}$ and $\pi'[0, i] = \pi[0, i]$ for all $\pi' \in \Pi$, it follows that for all $\ell \in J$, $(\mathcal{T}, \pi, i) \models \theta_{p,\ell}$. Thus, we obtain that $(\mathcal{T}, \pi, i) \models (\bigwedge_{\ell \in J} \theta_{p,\ell}) \wedge \langle\!\langle A\rangle\!\rangle(\bigvee_{\ell \in J} \theta_{f,\ell})$ which entails that $(\mathcal{T}, \pi, i) \models \psi_B$.

For the converse direction, assume that $(\mathcal{T}, \pi, i) \models \psi_B$. Hence, there are a non-empty subset $J \subseteq I$ and a strategy f_A for A such that the following holds, where Π is the non-empty set of ε-plays $\pi' \in out(\pi[0, i], f_A)$:

- for all $\ell \in J$, $(\mathcal{T}, \pi, i) \models \theta_{p,\ell}$;
- for all $\pi' \in \Pi$, there is $\ell \in J$ such that $(\mathcal{T}, \pi', i) \models \theta_{f,\ell}$.

Again since $\theta_{\ell,p}$ corresponds to a pure past LTL_p formula over $\overline{AP}$ and $\pi'[0,i] = \pi[0,i]$ for all $\pi' \in \Pi$, it follows that for all $\ell \in J$ and $\pi' \in \Pi$, $(\mathcal{T}, \pi', i) \models \theta_{p,\ell}$. It follows that $(\mathcal{T}, \pi, i) \models \langle\langle A \rangle\rangle \bigvee_{\ell \in J} (\theta_{p,\ell} \wedge \theta_{f,\ell})$ which entails that $(\mathcal{T}, \pi, i) \models \psi$. ◀

We now prove Lemma 11.

▶ **Lemma 11.** *Let φ be an ATL^*_{lp} path formula. Then, φ is congruent to a Boolean combination of ATL^* formulas and ATL^*_{lp} formulas that correspond to* pure past *LTL_p formulas over the set of propositions $AP \cup \mathcal{H}$, where $\mathcal{H}$ consists of basic ATL^* formulas.*

Proof. The proof is by induction on the nesting depth of the strategy quantifiers in φ.

Base case: in this case $BFL(\varphi) = \emptyset$, and the result directly follows from Lemma 17.

Inductive step: let $\langle\langle A' \rangle\rangle\psi \in BFL(\varphi)$. By the inductive hypothesis, the thesis holds for formula ψ. Hence, ψ is a congruent to an ATL^*_{lp} formula ψ' such that $BFL(\psi')$ consists of basic ATL^* formulas. By applying Lemma 17, $\langle\langle A' \rangle\rangle\psi$ is congruent to an ATL^*_{lp} formula, say ξ, such that $BFL(\xi)$ consists of basic ATL^* formulas. By replacing each occurrence of $\langle\langle A' \rangle\rangle\psi$ in φ with ξ, and repeating the procedure for all the formulas in $BFL(\varphi)$, we obtain an ATL^*_{lp} path formula θ which is congruent to φ (note that the congruence relation is closed under substitution) and such that $BFL(\theta)$ consists of basic ATL^* formulas. At this point we can apply Lemma 17 to formula θ proving the assertion. ◀

B Proof of Proposition 6

▶ **Proposition 6.** *Let ψ be a pure past LTL_p formula over $AP \cup B_\Phi$. Then, one can construct a safety DWA $\mathcal{D}_\psi = \langle 2^{AP \cup B_\Phi}, q_0, Q \cup \{q_0\}, \delta \rangle$ such that $Q \subseteq 2^{cl(\psi)}$, there is no transition leading to q_0, and the following holds for each* infinite *word w over $2^{AP \cup B_\Phi}$:*

- *let $q_0\, C_0\, C_1 \dots$ be the unique run of $\mathcal{D}_\psi$ over w. Then, for all $i \geq 0$ and $\theta \in cl(\psi)$, $\theta \in C_i$ if and only if $(w, i) \models \theta$.*

Proof. The safety DWA $\mathcal{D}_\psi = \langle 2^{AP \cup B_\Phi}, q_0, Q \cup \{q_0\}, \delta \rangle$ such that $Q \subseteq 2^{cl(\psi)}$ is defined as follows. Q is the set of *atoms* of ψ consisting of the maximal propositionally consistent subsets C of $cl(\psi)$. Formally, an atom C of ψ is a subset of $cl(\psi)$ satisfying the following:

- for each $\theta \in cl(\psi)$, $\theta \in C$ iff $\neg\theta \notin C$;
- for each $\theta_1 \vee \theta_2 \in cl(\psi)$, $\theta_1 \vee \theta_2 \in C$ iff $\{\theta_1, \theta_2\} \cap C \neq \emptyset$.

An atom is *initial* if

- for each $\mathsf{X}^-\theta \in cl(\psi)$, $\mathsf{X}^-\theta \notin C$;
- for each $\theta_1 \mathsf{U}^- \theta_2 \in cl(\psi)$, $\theta_1 \mathsf{U}^- \theta_2 \in C$ iff $\theta_2 \in C$.

The transition relation δ captures the semantics of the previous modalities, and the local fixpoint characterization of the since modalities. Formally, for each $\sigma \in 2^{AP \cup B_\Phi}$, $\delta(q_0, \sigma)$ consists of the *uniquely determined initial* atom C such that C *is consistent with the input symbol* σ, i.e., $C \cap (AP \cup B_\Phi) = \sigma \cap (AP \cup B_\Phi)$. Moreover, for each atom C of ψ and $\sigma \in 2^{AP \cup B_\Phi}$, $\delta(C, \sigma)$ consists of the *uniquely determined* atom C' of ψ such that C' is consistent with the input symbol σ, and:

- for each $\mathsf{X}^-\theta \in cl(\psi)$, $\mathsf{X}^-\theta \in C'$ iff $\theta \in C$;
- for each $\theta_1 \mathsf{U}^- \theta_2 \in cl(\psi)$, $\theta_1 \mathsf{U}^- \theta_2 \in C'$ iff either $\theta_2 \in C'$, or $\theta_1 \in C'$ and $\theta_1 \mathsf{U}^- \theta_2 \in C$.

By standard arguments (see [31]), given an infinite word w over $2^{AP \cup B_\Phi}$, for the unique run $\pi = q_0\, C_0\, C_1 \dots C_\ell$ of $\mathcal{D}_\psi$ over w, the following holds: for all $i \geq 0$ and $\theta \in cl(\psi)$, $\theta \in C_i$ if and only if $(w, i) \models \theta$. Hence, the result follows. ◀

C Proof of Lemma 13

▶ **Lemma 13** (Characterization of extended basic models of Φ)**.** *Let $\mathcal{T}$ be a CGT with labeling Lab over $AP \cup B_\Phi$ such that $Lab(\varepsilon)$ is a model of the propositional formula $[\Phi]_{\mathsf{LTL}_p}$. Then, $\mathcal{T}$ is an extended basic model of Φ* if and only if *for each basic subformula $\langle\langle A\rangle\rangle\psi \in B_\Phi$, $\mathcal{T}$ is both positively and negatively well-formed with respect to $\langle\langle A\rangle\rangle\psi$.*

Proof. Let $\mathcal{T}$ as in the statement of the lemma, and for a node x of $\mathcal{T}$, let ν_x be the track of $\mathcal{T}$ from the root to node x.

First assume that for each basic subformula $\langle\langle A\rangle\rangle\psi \in B_\Phi$, $\mathcal{T}$ is both positively and negatively well-formed with respect to $\langle\langle A\rangle\rangle\psi$. Fix a node x of $\mathcal{T}$ and $\langle\langle A\rangle\rangle\psi \in B_\Phi$. We need to show that $\langle\langle A\rangle\rangle\psi \in Lab(x)$ *iff* $(\mathcal{T}, x) \models \langle\langle A\rangle\rangle\psi$. The proof is by structural induction on the nesting depth of strategy quantifiers in ψ. We focus on the implication $\langle\langle A\rangle\rangle\psi \notin Lab(x) \Rightarrow (\mathcal{T}, x) \models \neg\langle\langle A\rangle\rangle\psi$ (the converse implication $\langle\langle A\rangle\rangle\psi \in Lab(x) \Rightarrow (\mathcal{T}, x) \models \langle\langle A\rangle\rangle\psi$ is similar) and assume that $\psi = \psi_1 \mathsf{U} \psi_2$ (the other cases being similar). Thus, let $\langle\langle A\rangle\rangle\psi \notin Lab(x)$. Since $\mathcal{T}$ is negatively well-formed with respect to $\langle\langle A\rangle\rangle\psi$, $\langle\langle A\rangle\rangle\psi \notin Lab(x)$, and $\psi = \psi_1 \mathsf{U} \psi_2$, by Definition 12 there exists a counter strategy f_A^c of $\mathcal{T}$ for A such that for all the outcomes $\pi \in out(\nu_x, f_A^c)$, the following holds, where $\mathcal{D}_{\psi_1}$ and $\mathcal{D}_{\psi_2}$ are the safety DWA of Proposition 6 associated with the pure past LTL_p formulas $[\psi_1]_{\mathsf{LTL}_p}$ and $[\psi_2]_{\mathsf{LTL}_p}$:

- (*) for all $j \geq |x|$ either $\neg\psi_2 \in \mathcal{D}_{\psi_2}(Lab(\pi[0, j]))$ or $\neg\psi_1 \in \mathcal{D}_{\psi_1}(Lab(\pi[0, k]))$ for some $k \in [|x|, j-1]$.

By applying Proposition 6 to the unique run of the DWA $\mathcal{D}_{\psi_1}$ (resp., $\mathcal{D}_{\psi_2}$) over the infinite word $Lab(\pi)$, by Condition (*) we deduce that

- (**) for all $j \geq |x|$ either $(Lab(\pi), j) \models_{\mathsf{LTL}} [\neg\psi_2]_{\mathsf{LTL}_p}$ or $(Lab(\pi), k) \models_{\mathsf{LTL}} [\neg\psi_1]_{\mathsf{LTL}_p}$ for some $k \in [|x|, j-1]$.

It follows that $(Lab(\pi), |x|) \models_{\mathsf{LTL}} [\neg\psi]_{\mathsf{LTL}_p}$. By the induction hypothesis, for each $\langle\langle A'\rangle\rangle\theta \in BFL(\psi)$ and position $j \geq 0$, we have that $\langle\langle A'\rangle\rangle\theta \in Lab(\pi(j))$ iff $(\mathcal{T}, \pi(j)) \models \langle\langle A'\rangle\rangle\theta$ (note that in the base step, $BFL(\psi) = \emptyset$, hence ψ in a pure past LTL_p formula over AP). Thus, by the semantics of ATL_{lp}, it follows that $(\mathcal{T}, \pi, |x|) \models \neg\psi$. Since π in an arbitrary outcome of the counter strategy f_A^c from the history ν_x, by Proposition 3, we conclude that $(\mathcal{T}, x) \models \neg\langle\langle A\rangle\rangle\psi$, and the result follows.

For the converse implication, assume that for each node x of $\mathcal{T}$ and basic subformula $\langle\langle A\rangle\rangle\psi \in B_\Phi$, $\langle\langle A\rangle\rangle\psi \in Lab(x)$ *iff* $(\mathcal{T}, x) \models \langle\langle A\rangle\rangle\psi$. Fix $\langle\langle A\rangle\rangle\psi \in B_\Phi$. We show that $\mathcal{T}$ is both positively and negatively well-formed with respect to $\langle\langle A\rangle\rangle\psi$. We focus on the negatively well-formedness condition and assume that $\psi = \psi_1 \mathsf{U} \psi_2$ (the other cases being similar). Thus, let x be a node of $\mathcal{T}$ such that $\langle\langle A\rangle\rangle\psi \notin Lab(x)$. By hypothesis, $(\mathcal{T}, x) \models \neg\langle\langle A\rangle\rangle\psi$. By Proposition 3, there exists a counter strategy f_A^c of $\mathcal{T}$ for A such that for all the outcomes $\pi \in out(\nu_x, f_A^c)$, it holds that $(\mathcal{T}, \pi, |x|) \models \neg\psi$, hence, $(Lab(\pi), |x|) \models_{\mathsf{LTL}} [\neg\psi]_{\mathsf{LTL}_p}$. Fix $\pi \in out(\nu_x, f_A^c)$. Let $q_0^1\, C_0^1\, C_1^1 \ldots$ (resp., $q_0^2\, C_0^2\, C_1^2 \ldots$) be the unique run of the safety DWA $\mathcal{D}_{\psi_1}$ (resp., $\mathcal{D}_{\psi_2}$). Since $(Lab(\pi), |x|) \models_{\mathsf{LTL}} [\neg\psi]_{\mathsf{LTL}_p}$ and $\psi = \psi_1 \mathsf{U} \psi_2$, by Proposition 6, it follows that for all $j \geq |x|$ either $\neg\psi_2 \in C_j^2$ or $\neg\psi_1 \in C_k^1$ for some $k \in [|x|, j-1]$.

Since, π and x are arbitrary, by Definition 12, we conclude that $\mathcal{T}$ is negatively well-formed with respect to $\langle\langle A\rangle\rangle\psi$. ◀

D Proof of Theorem 15

▶ **Theorem 15.** *Given an ATL_{lp} state formula Φ, one can build in single exponential time a Büchi ACG $\mathcal{A}'_\Phi$ over 2^{AP}, equipped with a* nondeterministic *satellite $\mathcal{U}'_\Phi$, that accepts the set of models of Φ. Moreover, $\mathcal{A}'_\Phi$ has $O(|\Phi|)$ states and $\mathcal{U}'_\Phi$ has $2^{O(|\Phi|)}$ states.*

Proof. Let $\mathcal{A}_\Phi$ and $\mathcal{U}_\Phi$ be the Büchi ACG and the deterministic satellite of Theorem 14 running on CGT with labeling over $2^{AP\cup B_\Phi}$. Starting from $\mathcal{A}_\Phi$ and $\mathcal{U}_\Phi$, we construct a new Büchi ACG $\mathcal{A}'_\Phi$ equipped with a nondeterministic satellite $\mathcal{U}'_\Phi$ over 2^{AP} accepting the set of models Φ. Intuitively, given an input CGT $\mathcal{T}$, $\mathcal{A}'_\Phi$ and $\mathcal{U}'_\Phi$ guess a B_Φ-labeling extension of the input $\mathcal{T}$ and check that such an extension is an extended basic model of Φ. In particular, we let the satellite $\mathcal{U}'_\Phi$ to guess the B_Φ-labeling extension and let the ACG $\mathcal{A}'_\Phi$ check the guess. Thus, $\mathcal{U}'_\Phi$ is obtained from $\mathcal{U}_\Phi$ by adding on top of the deterministic transition relation a guess of the subset of B_Φ to be read in the current node of the input CGT. Note that since an ACG is an alternating automaton, it is not to possible, in general, for an ACG to guess a labelling of the input, since different copies of the automaton which read the same input node may take different guesses. However, here, we exploit the crucial fact that for a CGT $\mathcal{T}$ over 2^{AP}, there is at most one B_Φ-labeling extension of $\mathcal{T}$ which is an extended basic model of Φ (Remark 4.3). Formally, let $\mathcal{A}_\Phi = \langle 2^{AP\cup B_\Phi}, q_0, Q, \delta, F\rangle$ and $\mathcal{U}_\Phi = \langle 2^{AP\cup B_\Phi}, s_0, S\cup\{s_0\}, \delta_S\rangle$. Then, $\mathcal{A}'_\Phi = \langle 2^{AP}, q_0, Q, \delta', F\rangle$ and $\mathcal{U}'_\Phi = \langle 2^{AP}, s_0, S\times 2^{B_\Phi}\cup\{s_0\}, \delta'_S\rangle$, where δ' and δ'_S are defined as follows:

- $\delta'_S(s_0,\sigma) = \bigcup_{\sigma'\in 2^{B_\Phi}} \delta_S(s_0,\sigma\cup\sigma')\times\{\sigma'\}$.
- $\delta'_S((s,\sigma'),\sigma) = \bigcup_{\sigma''\in 2^{B_\Phi}} \delta_S(s_0,\sigma\cup\sigma'')\times\{\sigma''\}$.
- $\delta'(q,\sigma,(s,\sigma')) = \delta(q,\sigma\cup\sigma',s)$.

We now prove that the construction is correct. Evidently, by construction and Theorem 14, it suffices to show that for each input $\mathcal{T}$, if $\mathcal{T}$ is accepted by $\mathcal{A}'_\Phi$, then $\mathcal{T}$ is a model of Φ. Let r be an accepting run of $\mathcal{A}'_\Phi$ over an input $\mathcal{T}$. Note that for each subformula ψ of Φ, since $[\psi]_{\mathsf{LTL}_p}$ is an LTL_p formula over $AP\cup BFL(\psi)$, the transition function $\delta_\psi(q,\sigma)$ of the safety DWA $\mathcal{D}_\psi$ of Proposition 6 is independent of the $B_\Phi\setminus BFL(\psi)$-part of σ. Thus, by a straightforward double induction on the nesting depth of strategy quantifiers in θ for a basic formula $\langle\langle A\rangle\rangle\theta\in B_\phi$ and the distance $|x|$ from the root of a node x of the input $\mathcal{T}$, and by construction and the proof of Theorem 14, the following holds:

- for all nodes x and copies of the nondeterministic satellite $\mathcal{U}'_\Phi$ in states (s',σ') and (s'',σ'') in the run r resulting on reading the node x, it holds that: $(s',\sigma')=(s'',\sigma'')$ and for all $\langle\langle A\rangle\rangle\theta\in B_\phi$, $\langle\langle A\rangle\rangle\theta\in\sigma'$ iff $(\mathcal{T},x)\models\langle\langle A\rangle\rangle\theta$.

Hence, the result follows. ◀

E Exptime-hardness of ATL$_{lp}$ model-checking

In this section, we establish the following result, where a two-player turn-based CGS is a CGS with two agents where each state is controlled by an agent.

▶ **Theorem 18.** *Model checking against ATL$_{lp}$ is* Exptime*–hard even for finite two-player turn-based CGS of fixed size.*

Theorem 18 is proved by a polynomial-time reduction from the acceptance problem for *linearly-bounded alternating* Turing Machines (TM) with a binary branching degree and with a fixed size, which is Exptime-complete [8]. In the rest of this section, we fix such a TM machine $\mathcal{M} = \langle\Sigma, Q, Q_\forall, Q_\exists, q_0, \delta, F\rangle$, where Σ is the input alphabet, Q is the finite set of states which is partitioned into $Q = Q_\forall\cup Q_\exists$, $Q_\exists$ (resp., $Q_\forall$) is the set of existential (resp., universal) states, q_0 is the initial state, $F\subseteq Q$ is the set of accepting states, and the transition function δ is a mapping $\delta: Q\times\Sigma\to(Q\times\Sigma\times\{L,R\})^2$. Moreover, we fix an input $\alpha\in\Sigma^*$ such that $|\alpha|\geq 1$ and consider the parameter $n=|\alpha|$.

Since $\mathcal{M}$ is linearly bounded, we can assume that $\mathcal{M}$ uses exactly n tape cells when started on the input α. Hence, a TM configuration (of $\mathcal{M}$ over α) is a word $C = \eta\cdot(q,\sigma)\cdot\eta'\in$

$\Sigma^* \cdot (Q \times \Sigma) \cdot \Sigma^*$ of length exactly n denoting that the tape content is $\eta \cdot \sigma \cdot \eta'$, the current state is q, and the tape head is at position $|\eta| + 1$. From configuration C, the machine $\mathcal{M}$ nondeterministically chooses a triple (q', σ', dir) in $\delta(q, \sigma) = \langle (q_l, \sigma_l, dir_l), (q_r, \sigma_r, dir_r) \rangle$, and then moves to state q', writes σ' in the current tape cell, and its tape head moves one cell to the left or to the right, according to dir. We denote by $succ_l(C)$ and $succ_r(C)$ the successors of C obtained by choosing respectively the left and the right triple in $\langle (q_l, \sigma_l, dir_l), (q_r, \sigma_r, dir_r) \rangle$. The configuration C is accepting (resp., universal, resp., existential) if the associated state q is in F (resp., in $Q_\forall$, resp., in $Q_\exists$). A (finite) computation tree of $\mathcal{M}$ over α is a finite tree in which each node is labeled by a configuration. The root of the tree corresponds to the initial configuration C_α given by $(q_0, \alpha(0))\alpha(1) \ldots \alpha(n-1)$. An *internal* node that is labeled by a universal configuration C has two children, corresponding to $succ_l(C)$ and $succ_r(C)$, while an internal node labeled by an existential configuration C has a single child, corresponding to either $succ_l(C)$ or $succ_r(C)$. The tree is accepting iff each its leaf is labeled by an accepting configuration. The input α is *accepted* by $\mathcal{M}$ iff there is an accepting computation tree of $\mathcal{M}$ over α. We prove the following result from which Theorem 18 directly follows.

▶ **Theorem 19.** *One can construct, in time polynomial in n and the size of $\mathcal{M}$, a finite turn-based* CGS *$\mathcal{G}$ and an* ATL_{lp} *state formula φ over the set of agents $Ag = \{ag_\exists, ag_\forall\}$ such that $\mathcal{M}$ accepts α* iff *$Unw(\mathcal{G}) \models \varphi$. Moreover, the size of $\mathcal{G}$ depends only on the size of $\mathcal{M}$.*

In order to prove Theorem 19, we first define a suitable encoding of the TM computations of $\mathcal{M}$ over α. Formally, we exploit the set AP of atomic propositions given by $AP := \Sigma \cup Q \times \Sigma \times \{\forall, \exists, l, r, acc\}$. A TM configuration $C = u_1 u_2 \ldots u_n$ is encoded by words w_C over 2^{AP} of the form $w_C = \{tag_1\}\{u_1\} \ldots \{u_k\}\{tag_2\}$, where $tag_1 \in \{l, r\}$, $tag_2 = acc$ if C is an accepting configuration, $tag_2 = \exists$ if C is a non-accepting existential configuration, and $tag_2 = \forall$ otherwise. The symbols l and r are used to mark a left and a right TM successor, respectively. We also use the symbol l to mark the initial configuration C_α. A sequence $w_{C_1} \cdot \ldots \cdot w_{C_p}$ of TM configuration codes is *good* if the following holds:

- C_1 is the initial configuration and w_{C_1} is marked by symbol l;
- C_p is accepting;
- for each $1 \leq i < p$, either $w_{C_{i+1}}$ is marked by symbol l and $C_{i+1} = succ_l(C_i)$, or $w_{C_{i+1}}$ is marked by symbol r and $C_{i+1} = succ_r(C_i)$.

Now, we prove Theorem 19. The finite turn-based CGS $\mathcal{G}$ over the set of agents $Ag = \{ag_\exists, ag_\forall\}$ in Theorem 19 is defined as follows:

- the set of states of $\mathcal{G}$ is AP and the initial state is l;
- the label of each state $p \in AP$ is $\{p\}$;
- state $\forall$ is controlled by agent $ag_\forall$, and each other state is controlled by agent $ag_\exists$;
- state $\forall$ has as successors the states l and r, state acc has as successor itself, while each other state has as successors the whole set of states.

Note that the CGS $\mathcal{G}$ is independent of n. By construction, the following hold.

Claim. $\mathcal{M}$ accepts α iff there exists a strategy $f_\exists$ of agent $ag_\exists$ in $Unw(\mathcal{G})$ such that for all ε-plays π consistent with $f_\exists$, the label of π is an infinite word over 2^{AP} of the form $w \cdot \{acc\}^\omega$ such that w is a *good* sequence of TM configuration codes.

Thus, the ATL_{lp} formula φ satisfying the statement of Theorem 19 is given by $\varphi := \langle\langle ag_\exists \rangle\rangle \mathsf{F}(acc \wedge \psi_{good})$, where ψ_{good} is a pure past LTL_p formula of size $O(n^2)$ capturing the good sequences of TM configuration codes. The construction of ψ_{good} is standard and, therefore, we omit further details.