How to Manage a Budget with ATL^{+}

Stéphane Demri[1] , **Raine Rönnholm**[2]
[1]LMF, CNRS, ENS Paris-Saclay, University Paris-Saclay, France
[2]Faculty of Information Technology and Communication Sciences, Tampere University, Finland
demri@lmf.cnrs.fr, raine.ronnholm@tuni.fi

Abstract

We study ATL^{+} enriched with one resource (written ATL$^{+}(1)$) extending ATL^{+} with the possibility to manage a budget. We propose a game-theoretic semantics via the introduction of two evaluation games so that the compositional semantics is captured by strategies in the games. We show that the model-checking problem for ATL$^{+}(1)$ is in PSPACE and we identify fragments in PTIME. By-products of our investigations include a simplified PSPACE decision procedure for resource-free ATL^{+} based on small strategy skeletons, the synthesis of constraints in ATL$^{+}(1)$ with parameters and a PSPACE bound to solve a close energy game problem with one counter and objectives of temporal depth one.

1 Introduction

ATL-like logics with resources. Reasoning about the ability of autonomous agents to achieve a goal, possibly in cooperation or against other agents, is essential to reason about knowledge and many logical formalisms have been introduced to perform such tasks, including alternating-time temporal logic ATL, see e.g. (Alur, Henzinger, and Kupferman 2002) and strategy logic SL, see e.g. (Chatterjee, Henzinger, and Piterman 2010). Since the seminal work (Alur, Henzinger, and Kupferman 2002) about ATL-like logics to reason about concurrent game structures (CGS), alternating-time temporal logics have been the subject of numerous investigations related to expressive power (see e.g. (Laroussinie, Markey, and Oreiby 2008)), game-based semantics (see e.g. (Goranko, Kuusisto, and Rönnholm 2018; Goranko, Kuusisto, and Rönnholm 2021)), computational complexity (see e.g. (Bulling and Jamroga 2010; Goranko and Vester 2014; Goranko, Kuusisto, and Rönnholm 2021)) and reasoning with resources (see e.g. (Bulling and Farwer 2009; Alechina et al. 2017; Bulling and Goranko 2022)), to quote a few examples. Adding resources to ATL-like logics can be done in many ways, see e.g. (Alechina et al. 2009; Bulling and Farwer 2010; Alechina et al. 2017; Bulling and Goranko 2022) (see also (Laroussinie, Markey, and Oreiby 2006; Vester 2014)). This is a natural framework in which each action done by some agent either consumes or produces resources. Decidability and complexity issues are considered in (Bulling and Farwer 2010; Alechina et al. 2015; Alechina et al. 2017; Alechina et al. 2018), some of them related to energy games, see e.g. (Colcombet et al. 2017), and to games on vector addition systems with states (VASS), see e.g. (Brázdil, Jancar, and Kucera 2010). In this paper, we are mainly interested in ATL^{+}, following investigations in (Bulling and Jamroga 2010; Cerrito, David, and Goranko 2015; Goranko, Kuusisto, and Rönnholm 2017; Cerrito 2021), but enriched with one resource.

Our motivations. The logic ATL^{+} extends the more popular ATL in the same way CTL^{+} extends CTL: the objectives for the coalitions are Boolean combinations of LTL formulae of temporal depth one. ATL^{+} is a non-trivial extension of ATL as its model-checking problem is PSPACE-complete (see e.g. (Alur and La Torre 2001; Bulling and Jamroga 2010) for the PSPACE-hardness and (Goranko, Kuusisto, and Rönnholm 2021) for the PSPACE-membership) whereas only in PTIME for ATL (Alur, Henzinger, and Kupferman 2002). The game-theoretic semantics for ATL^{+} formulae designed in (Goranko, Kuusisto, and Rönnholm 2021) provides a neat characterisation for the winning strategies witnessing the satisfaction of formulae. Our main motivation in this work is to provide a game-theoretic semantics for resource-aware ATL-like logics. Moreover, we aim at characterising the complexity of model-checking. To do so, we consider ATL^{+} enriched with one resource (ATL$^{+}(1)$). A version of ATL enriched with one resource is solved in PSPACE in (Alechina et al. 2017, Theorem 2). We already know that the model-checking problem for ATL$^{+}(1)$ is PSPACE-hard (inherited from ATL^{+}) and for a variant of ATL$^{+}(1)$, in EXPTIME (Belardinelli and Demri 2021, Theorem 14). The fact that we restrict ourselves to a single resource is motivated by the need to elaborate the first steps to design game-theoretic semantics for resource-aware ATL-like logics. Besides, bounding the number of counters is a natural approach to study decision problems on VASS, see e.g. (Czerwinski et al. 2020; Leroux and Schmitz 2019), or to analyse energy games, see e.g. (Jurdziński, Lazić, and Schmitz 2015). Energy games with one resource are also investigated, see e.g. (Bouyer et al. 2008; Chatterjee, Doyen, and Henzinger 2017).

Our contributions. We introduce the logic ATL$^{+}(1)$ following first principles from (Alechina et al. 2017). We keep to the minimum the conditions about the resource values, typically non-negativity for winning strategies (no compulsory idle action, no proponent restriction condition). Our intention remains to define a simple version of ATL^+ with one resource, although our results can be adapted to variants with more conditions (dropping such conditions herein is not due to hidden technical difficulties).

We design two evaluation games, along the lines of (Goranko, Kuusisto, and Rönnholm 2018; Goranko, Kuusisto, and Rönnholm 2021) (see Section 3), that are shown equivalent to the compositional semantics for $\mathsf{ATL}^+(1)$. The games are played between two players, Eloise (**E**) and Abelard (**A**). Intuitively, **E** is trying to verify that a state from a CGS satisfies a formula in $\mathsf{ATL}^+(1)$. The first evaluation game admits a transition game (subgame of the evaluation game) in which loop checking is performed leading to specific winning positions for **E**. Each position of a play carries a resource value and winning conditions depend on the negativity status of the current resource value and/or of the visited loops. Once the equivalence with compositional semantics is shown, a PSPACE upper bound for the model-checking problem for $\mathsf{ATL}^+(1)$ (written MC($\mathsf{ATL}^+(1)$)) can be easily concluded. The other evaluation game is parameterised by a threshold value $\beta > 0$ for which any resource value strictly above β is immediately truncated to β. Consequently, if β is too small, truncation may effectively change the existence of a winning strategy for **E**. This version of the game is shown equivalent to the compositional semantics for β large enough. For each β, existence of a winning strategy for **E** is equivalent to existence of a winning strategy in a Büchi game, for which we can then use all the nice properties, see e.g (Chatterjee, Henzinger, and Piterman 2008). This is the way we design PTIME fragments of MC($\mathsf{ATL}^+(1)$). Our games are inspired from (Goranko, Kuusisto, and Rönnholm 2021) but are sometimes more complex due to the presence of a resource value in positions; still we provide several technical simplifications.

In Section 5.2, we define a parameterised version of $\mathsf{ATL}^+(1)$ for which we explain how quantifier-free Presburger formulae can be synthetised in polynomial space. In such a parameterised version, the strategy modality $\langle\langle A^b \rangle\rangle \Psi$ from $\mathsf{ATL}^+(1)$ where $b \in \mathbb{N}$ (initial budget) and Ψ is an $\mathsf{ATL}^+(1)$ path formula is replaced by $\langle\langle A^{\mathtt{x}} \rangle\rangle \Psi$ where $\mathtt{x}$ is a variable. The synthesis problem that we solve consists in building an arithmetical constraint that captures exactly the values for initial budgets that make true a given parameterised formula on a given state. Finally, in Section 5.3, we prove that an energy game problem with LTL objectives of temporal depth one can be reduced in logspace to MC($\mathsf{ATL}^+(1)$), leading to its PSPACE upper bound.

2 Preliminaries

2.1 ATL^+ Enriched With One Resource Type

Below, we write $\mathsf{ATL}^+(1)$ to denote the logic ATL^+ (see e.g. (Bulling and Jamroga 2010; Cerrito, David, and Goranko 2015)) augmented with one resource. Its models are concurrent game structures in which each action has a weight defined as an integer. The weight is subject to several interpretations, for example to be understood as a cost. A *concurrent game structure with one resource* (in short, a CGS) is a tuple $M = \langle Ag, S, Act, act, wf, \delta, L \rangle$ such that:

- Ag (resp. S, Act) is a finite and non-empty set of *agents* (resp. *states*, *actions*).
- $act : S \times Ag \to (\mathcal{P}(Act) \setminus \{\emptyset\})$ is the *protocol function* assigning to each pair (s, a) the set of available actions for the agent a at the state s. A *joint action* $f : Ag \to Act$ *at the state* s is a map such that for all $a \in Ag$, $f(a) \in act(s, a)$. Restrictions $g : A \to Act$ with $A \subseteq Ag$ of joint actions, are also called *joint actions w.r.t.* A *at* s. The set of joint actions w.r.t to A at s is written $D_A(s)$.
- $wf : S \times Ag \times Act \to \mathbb{Z}$ is a (partial) *weight function* such that for all $s \in S$, $a \in Ag$, and $\mathtt{a} \in Act$, $wf(s, a, \mathtt{a})$ is defined exactly when $\mathtt{a} \in act(s, a)$.
- $\delta : S \times (Ag \to Act) \to S$ is a (partial) *transition function* with $\delta(s, f)$ defined whenever f is a joint action at s.
- $L : \mathrm{AP} \to \wp(S)$ is the *labelling function*, where AP is a set of propositional variables.

We write $\Delta(M)$ to denote $\max\{|D_{Ag}(s)| : s \in S\}$. Observe that $\Delta(M) \leq |Act|^{|Ag|}$ and $\Delta(M)$ corresponds to the maximal number of outgoing transitions from a state when M is represented as a labelled graph. We also write $||M||_\infty$ to denote $\max\{|wf(s, a, \mathtt{a})| : s \in S,\ a \in Ag,\ \mathtt{a} \in Act\}$. Since we shall deal with complexity issues, the integers are encoded with a binary representation. We adopt a reasonably succinct encoding for M such that its size, written $|M|$, is polynomial in $|Ag| + |S| + |Act| + log(||M||_\infty) + \Delta(M)$.

Given joint actions f and g, we write $g \sqsubseteq f$ if $dom(g) \subseteq dom(f)$, and for every agent $a \in dom(g)$, $f(a) = g(a)$. For a joint action $g \in D_A(s)$, $out(s, g)$ is the set of *immediate outcomes*: $out(s, g) \stackrel{\text{def}}{=} \{\delta(s, f) \mid \text{for some } f \in D_{Ag}(s),\ g \sqsubseteq f\}$. Given a coalition $A \subseteq Ag$, the coalition $(Ag \setminus A)$, also written $\overline{A}$, is called the *opposing coalition*. We write $g \oplus \overline{g}$ to denote the joint action in $D_{Ag}(s)$ when $g \in D_A(s)$ and $\overline{g} \in D_{\overline{A}}(s)$. The *weight* of a transition from s by $f \in D_{Ag}(s)$ is defined as $wf(s, f) \stackrel{\text{def}}{=} \sum_{a \in Ag} wf(s, a, f(a)) \in \mathbb{Z}$.

A *computation* λ is a finite or infinite sequence $s_0 \xrightarrow{f_0} s_1 \xrightarrow{f_1} s_2 \ldots$ such that for all $0 \leq i < |\lambda| - 1$ we have $s_{i+1} = \delta(s_i, f_i)$. A computation can be also defined as a sequence $s_0 s_1 s_2 \cdots$ (obtained by removing the joint actions) assuming that for all $0 \leq i < |\lambda| - 1$, there is some f_i such that $s_{i+1} = \delta(s_i, f_i)$. When we use the notation $s_0 s_1 s_2 \cdots$, we assume that the joint actions $f_0 f_1 f_2 \cdots$ are known. Moreover, if $\lambda = s_0 \cdots s_K$, we write $\lambda(i)$ to denote s_i and $\lambda[i, j]$ for the sequence $s_i \cdots s_j$. A similar notation is used with infinite computations.

A *strategy* σ *for the coalition* A is a map from the set of finite computations to the set of joint actions of A such that

$$\sigma(s_0 \xrightarrow{f_0} s_1 \ldots s_{n-1} \xrightarrow{f_{n-1}} s_n) \in D_A(s_n).$$

Positional strategies are such that $\sigma(s_0 \cdots s_n)$ only depends on s_n. A computation $\lambda = s_0 \xrightarrow{f_0} s_1 \xrightarrow{f_1} s_2 \ldots$ *respects the strategy* σ iff for all $i < |\lambda| - 1$, $s_{i+1} \in out(s_i, \sigma(s_0 \xrightarrow{f_0} s_1 \ldots s_{i-1} \xrightarrow{f_{i-1}} s_i))$. The set of all the infinite computations starting from s and respecting σ is written $comp(s, \sigma)$.

Given $b \in \mathbb{N}$ and an infinite computation $\lambda = s_0 \xrightarrow{f_0} s_1 \xrightarrow{f_1}$

$s_2 \ldots$, let the *resource availability* v_i *at step* $i \in \mathbb{N}$ be defined as: $v_0 \stackrel{\text{def}}{=} b$ and for all $i \geq 0$, $v_{i+1} \stackrel{\text{def}}{=} wf(s_i, f_i) + v_i$. In short, each v_i is the accumulated weight at position i, assuming that the initial budget is b. The infinite computation λ is *b-consistent* iff for all $i \in \mathbb{N}$, we have $v_i \geq 0$. A strategy σ is a *b-strategy w.r.t. s* iff all the computations in $comp(s, \sigma)$ are b-consistent. A *state-resource history* π is a sequence $((s_0, v_0), \ldots, (s_k, v_k))$ for some $k \geq 0$.

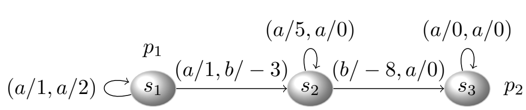
The models of the logic $\mathsf{ATL}^+(Ag, 1)$ (parameterised by the set of agents Ag) are CGSs containing the set of agents Ag. When defining formulae we make explicit the set of agents as (finite) coalitions occur in formulae. The *state formulae* ϕ and *path formulae* Ψ in the logic $\mathsf{ATL}^+(Ag, 1)$ are built according to the following grammar:

$$\phi ::= p \mid \neg\phi \mid \phi \vee \phi \mid \langle\langle A^b \rangle\rangle \Psi$$
$$\Psi ::= \phi \mid \neg\Psi \mid \Psi \vee \Psi \mid \mathsf{X}\,\phi \mid \phi \,\mathsf{U}\, \phi,$$

where $p \in \text{AP}$, $A \subseteq Ag$, and $b \in \mathbb{N}$. The peculiarity of the path formulae Ψ rests on the closure under Boolean connectives but no nesting of temporal connectives X and U is allowed, similarly to syntactic constraints for ATL^+ and CTL^+, see also the unnested-path-formula fragment in (Chatterjee, Henzinger, and Piterman 2010, Section 3.3). The standard Boolean connectives $\wedge$ and $\rightarrow$ are used as shorthands as well as for the temporal operators G and F. For instance, $\mathsf{G}\,\phi \stackrel{\text{def}}{=} \neg(\top \,\mathsf{U}\, \neg\phi)$. Below, the formula $\langle\langle A^b \rangle\rangle \Psi$ can be read as "the coalition A has a joint strategy implementable with initial budget b such that all the computations respecting that strategy satisfy the objective Ψ". By *$\mathsf{ATL}^+(Ag, 1)$ formulae*, by default we understand the state formulae. The satisfaction relation $\models$ is defined as follows.

$M, s \models p$	iff	$s \in L(p)$
$M, s \models \neg\phi$	iff	$M, s \not\models \phi$
$M, s \models \phi_1 \vee \phi_2$	iff	$M, s \models \phi_1$ or $M, s \models \phi_2$
$M, s \models \langle\langle A^b \rangle\rangle \Psi$	iff	for some b-strategy σ w.r.t. s for A, for all $\lambda \in comp(s, \sigma)$, $M, \lambda \models \Psi$
$M, \lambda \models \phi$	iff	$M, \lambda(0) \models \phi$
$M, \lambda \models \neg\Psi$	iff	$M, \lambda \not\models \Psi$
$M, \lambda \models \Psi_1 \vee \Psi_2$	iff	$M, \lambda \models \Psi_1$ or $M, \lambda \models \Psi_2$
$M, \lambda \models \mathsf{X}\,\phi$	iff	$M, \lambda(1) \models \phi$
$M, \lambda \models \phi_1 \,\mathsf{U}\, \phi_2$	iff	for some $i \geq 0$, $M, \lambda(i) \models \phi_2$, and for all $0 \leq j < i$, $M, \lambda(j) \models \phi_1$

Below, we present a CGS M with two agents (our running example), the transitions are labelled by pairs of actions with the respective weights. For instance, the total weight of the transition from s_1 to s_2 is -2. One can show that $M, s_1 \models \langle\langle \{1\}^3 \rangle\rangle (\mathsf{G}\,p_1 \vee \mathsf{F}\,p_2)$ but there is no positional strategy for agent 1 that witnesses the satisfaction of the formula.



Let M be a finite CGS with $Ag = [1, k]$ for some $k \geq 1$, s be a state in M and ϕ be a formula in $\mathsf{ATL}^+(Ag, 1)$ built over Ag. The *model-checking problem for $\mathsf{ATL}^+(1)$*, written $\text{MC}(\mathsf{ATL}^+(1))$, amounts deciding whether $M, s \models \phi$. This is the main problem we investigate in this work.

Given a path formula Ψ, we write $atoms(\Psi)$ to denote its finite set of *temporal atoms* made of

1. the maximal state formulae occurring in Ψ not in the scope of a temporal connective, and
2. the path formulae of the form either $\mathsf{X}\,\phi_1$ or $\phi_1 \,\mathsf{U}\, \phi_2$ in Ψ but none of them in the scope of a strategy modality.

With $\Psi_0 = (\langle\langle A^b \rangle\rangle \Psi' \vee p) \wedge (\mathsf{X}\,\phi_1 \wedge (\phi_3 \,\mathsf{U} \langle\langle A^b \rangle\rangle \Psi''))$, $atoms(\Psi_0) = \{\langle\langle A^b \rangle\rangle \Psi' \vee p, \mathsf{X}\,\phi_1, \phi_3 \,\mathsf{U} \langle\langle A^b \rangle\rangle \Psi''\}$. We write $msf(\Psi)$ to denote the maximal state formulae obtained from the temporal atoms in $atoms(\Psi)$. For instance, $msf(\Psi_0) = \{\langle\langle A^b \rangle\rangle \Psi' \vee p, \phi_1, \phi_3, \langle\langle A^b \rangle\rangle \Psi''\}$. The *temporal width* of Ψ, written $tw(\Psi)$, is defined as $|atoms(\Psi)|$. The *temporal width* of a state formula ϕ, written $tw(\phi)$, is defined as $\max\{tw(\Psi) \mid \text{ path formula } \Psi \text{ occurs in } \phi\}$. Given a state formula ϕ, we write $subf(\phi)$ its set of (state) subformulae (built as usual).

$\text{MC}(\mathsf{ATL}^+(1))$ can be solved with a standard labelling algorithm as soon as we know how to decide $M, s \models \langle\langle A^b \rangle\rangle \Psi$. The algorithm determines bottom-up on the structure of the subformulae ψ of ϕ which states in M satisfy ψ. Assuming that the satisfaction of formulae in $msf(\Psi)$ is already known, the verification of $M, s \models \langle\langle A^b \rangle\rangle \Psi$ is the key part of the algorithm. If checking $M, s \models \langle\langle A^b \rangle\rangle \Psi$ can be done in PSPACE using an oracle for the formulae in $msf(\Psi)$, then $\text{MC}(\mathsf{ATL}^+(1))$ is in PSPACE. Later on, we use the same reasoning with PTIME for fragments of $\text{MC}(\mathsf{ATL}^+(1))$. This approach does not work, if the formulae were interpreted on pairs $(s, v) \in S \times \mathbb{N}$, see e.g. (Vester 2014, Section 4).

2.2 Variants

ATL^+ corresponds to $\mathsf{ATL}^+(1)$ with all weights and budgets equal to zero. $\text{MC}(\mathsf{ATL}^+)$ is PSPACE-hard (Bulling and Jamroga 2010) and in PSPACE (Goranko, Kuusisto, and Rönnholm 2021). Herein we focus on $\text{MC}(\mathsf{ATL}^+(1))$ but our developments can be adapted to variant logics. For instance, we write $\mathsf{ATL}^+_p(1)$ for the variant of $\mathsf{ATL}^+(1)$ in which the *proponent restriction condition* is assumed: the weight of a transition from s by $f \in D_{Ag}(s)$ when the proponent coalition is A is defined as $wf(s, f) = \sum_{a \in A} wf(s, a, f(a))$. Only the agents in A are taken into account for computing weights, see the logics from (Alechina et al. 2017; Alechina et al. 2018). Our results for $\text{MC}(\mathsf{ATL}^+(1))$ can be adapted to $\text{MC}(\mathsf{ATL}^+_p(1))$.

The above variant rests on a slightly different semantics, but it is also possible to define relevant syntactic fragments, such as $\mathsf{ATL}(1)$, for which the path formulae are those from $\mathsf{ATL}^+(1)$ but restricted to $\Psi ::= \mathsf{X}\,\phi \mid \mathsf{G}\,\phi \mid \phi \,\mathsf{U}\, \phi$.

3 Evaluation Games with Resources

In (Goranko, Kuusisto, and Rönnholm 2021), a game-theoretic semantics is designed for ATL^+. Not only this leads to a semantics that is alternative to the compositional semantics but also the PSPACE upper bound for $\text{MC}(\mathsf{ATL}^+)$ is shown (correcting a flaw in (Bulling and Jamroga 2010, Theorem 4), see also an attempt in (Wang, Schewe, and Huang 2015)). Fragments for which the model-checking problem is in PTIME are also designed there (the temporal width is bounded as in ATL). Assuming that $\mathcal{G}(M, s, \phi)$ denotes the evaluation game for ATL^+ in (Goranko, Kuusisto, and Rönnholm 2021), it is shown that $M, s \models \phi$ iff the player **E** has a winning strategy for $\mathcal{G}(M, s, \phi)$.

In this section, we introduce two evaluation games related to $\mathsf{ATL}^+(1)$ extending the games for ATL^+. We perform several simplifications (for instance, our games do not deal with so-called *timer* and *seeker role*) but as a new feature, each position of the plays needs to carry a resource value. Each evaluation game comes with its version of a so-called transition game understood as a subgame of the evaluation game (related to the satisfaction of formulae $\langle\!\langle A^b \rangle\!\rangle \Psi$). We distinguish the transition game with loop checking from the resource-capped transition game. In the winning conditions, the resource value is taken into account in several ways. Firstly, a negative resource value most often is winning for **A** as a b-strategy does not admit computations leading to negative resource values. Secondly, in the version of the transition game with loop checking, non-negative loops can lead to a winning position for **E** only if it ends the play and the truth function satisfies the path formula Ψ. By contrast, in the resource-capped transition game, no loop checking is performed and the current resource value is truncated to a value β that is a parameter of the whole game.

3.1 Loop-Minimal Evaluation Game

Let M be a CGS, s_{in} be a state in M and ϕ_{in} be an $\mathsf{ATL}^+(1)$-formula. The *(loop-minimal) evaluation game* $\mathcal{G}_{\text{LM}}(M, s_{in}, \phi_{in})$ is played between two players *Eloise* (**E**) and *Abelard* (**A**). The *positions* of the game are of the form $(\mathbf{V}, s, \phi)$, where $\mathbf{V} \in \{\mathbf{E}, \mathbf{A}\}$ is the (current) *verifier*, $s \in S$ and $\phi \in subf(\phi_{in})$. The *opponent* of a player **P** is denoted by $\overline{\mathbf{P}}$ (that is, $\{\overline{\mathbf{P}}\} = \{\mathbf{E}, \mathbf{A}\} \setminus \{\mathbf{P}\}$). The opponent $\overline{\mathbf{V}}$ of the verifier can be also called the *falsifier*.

The game begins from the *initial position* $(\mathbf{E}, s_{in}, \phi_{in})$, where intuitively "Eloise is trying to verify that ϕ_{in} is true at s_{in}". The game is played according to the following rules.

1. In $(\mathbf{V}, s, \phi_1 \vee \phi_2)$, the verifier **V** selects whether the next position is $(\mathbf{V}, s, \phi_1)$ or $(\mathbf{V}, s, \phi_2)$.
2. In $(\mathbf{V}, s, \neg\phi)$, the game proceeds to the position $(\overline{\mathbf{V}}, s, \phi)$. A negation swaps the roles of the verifier and the falsifier.
3. In $(\mathbf{V}, s, \langle\!\langle A^b \rangle\!\rangle \Psi)$, the game enters the *transition game* $\mathcal{TG}_{\text{LM}}(\mathbf{V}, s, A, b, \Psi)$ which is defined below.
4. In $(\mathbf{V}, s, p)$ the evaluation game ends and the verifier **V** wins if and only if $s \in L(p)$ (else the falsifier $\overline{\mathbf{V}}$ wins).

An essential part of transition games is the use of *truth functions for a path formula* Ψ which are mappings $T : atoms(\Psi) \to \{\top, \bot, open\}$. The value $\top$ refers to "true", $\bot$ refers to "false". If $T(\Phi) = open$, then the satisfaction of Φ remains undecided but if the truth function does not evolve anymore, then *open* refers to "false". However, a truth function T can be updated during the game so that some values that are still *open* are remapped to either $\top$ or $\bot$. Since the values $\top$ and $\bot$ cannot be modified any further, there can be only finitely many such updates ($atoms(\Psi)$ is finite). The truth of Φ with respect to T ($atoms(\Phi) \subseteq atoms(\Psi)$), denoted by $T \models \Phi$, is defined as follows:

- $T \models \Phi$ iff $T(\Phi) = \top$, when $\Phi \in atoms(\Psi)$.
- $T \models \neg\Phi$ iff $T \not\models \Phi$; $T \models \Phi_1 \vee \Phi_2$ iff $T \models \Phi_1$ or $T \models \Phi_2$.

E.g., $[\top \mathsf{U} \neg p_1 \mapsto open, \top \mathsf{U} p_2 \mapsto \bot] \models \mathsf{G} p_1 \vee \mathsf{F} p_2$.

Below, we define the transition games which can be seen as subgames within evaluation games for verifying strategic subformulae $\langle\!\langle A^b \rangle\!\rangle \Psi$. Note that the outcome of a transition game may either be: an immediate win for one of the players; or position from which the evaluation game is continued with the rules above (possibly leading to new transition games for subformulae). The *transition game* $\mathcal{TG}_{\text{LM}}(\mathbf{V}, s, A, b, \Psi)$ is defined as follows. The positions of the game are triples of the form (π, T, l^+).

- π is a state-resource history $((s_0, b_0), \ldots, (s_k, b_k))$. The last state s_k in π is called the *current state* and b_k is called the *current resource value*; T is a truth function for Ψ.
- $l^+ \in \{0, 1\}$ is bit indicating whether a "positive loop" has been observed during the transition game (see (ii) below).

The initial position of $\mathcal{TG}_{\text{LM}}(\mathbf{V}, s, A, b, \Psi)$ is $(\pi_0, T_{open}, 0)$, where $\pi_0 = ((s, b))$ and T_{open} maps all temporal atoms to *open*. Then the game is played by iterating phases (i) → (ii) → (iii) → (iv) → (i) → ⋯.

(i) Verification phase. In this phase the players may attempt to *verify* or *falsify* the temporal atoms in $\Phi \in atoms(\Psi)$. Only Φ such that $T(\Phi) = open$ may be verified (resp. falsified) by mapping them to $\top$ (resp. $\bot$).

First, either player $\mathbf{P} \in \{\mathbf{E}, \mathbf{A}\}$ may attempt to verify formulae $\phi_1 \mathsf{U} \phi_2 \in atoms(\Psi)$, by claiming that ϕ_2 is true at current s_k. If the opponent $\overline{\mathbf{P}}$ does not object this claim, then T is updated by mapping $\phi_1 \mathsf{U} \phi_2$ to $\top$. If $\overline{\mathbf{P}}$ objects, then the claim is evaluated by exiting the transition game and resuming to the evaluation game from the position $(\mathbf{P}, s_k, \phi_2)$.

Then, similarly either player **P** may attempt to falsify $\phi_1 \mathsf{U} \phi_2 \in atoms(\Psi)$ by claiming that ϕ_1 and ϕ_2 are false at s_k. If $\overline{\mathbf{P}}$ does not object, then $\phi_1 \mathsf{U} \phi_2$ is mapped to $\bot$ by T; and if $\overline{\mathbf{P}}$ objects, the evaluation game continues from $(\overline{\mathbf{P}}, s_k, \phi_1)$ or $(\overline{\mathbf{P}}, s_k, \phi_2)$ depending on $\overline{\mathbf{P}}$'s choice.

The state formulae ϕ and X-formulae $\mathsf{X}\,\phi$ in $atoms(\Psi)$ can be both verified by **P** by claiming that ϕ is true at s_k; if the opponent objects, the evaluation game continues from $(\mathbf{P}, s_k, \phi)$. However, ϕ (resp. $\mathsf{X}\,\phi$) may only be verified in the *1st* (resp. *2nd*) verification phase of the transition game.

Resetting the history π: If at least one modification is done to T during (i), then π is initialized to $\pi := ((s_k, b_k))$.

Steps (ii) and (iii) below are related to loop detection. In (ii), the parameters are updated but the play continues. By contrast, in (iii), a loop is detected and the play terminates with a winner. A loop is detected on $\pi = ((s_0, b_0), \ldots, (s_k, b_k))$ if there is $j < k$ such that $s_j = s_k$. One can check that (ii) and (iii) cover all the possible cases.

(ii) First positive loop. If a loop is made in $\pi = ((s_0, b_0), \ldots, (s_k, b_k))$ with $j < k$, $s_j = s_k$, $b_j < b_k$ and $l^+ = 0$, then l^+ is set to 1 and π is reset to $((s_k, b_k))$.

(iii) Loop check phase (terminal position). If a loop is made in $\pi = ((s_0, b_0), \ldots, (s_k, b_k))$ with $j < k$ (excluding l^+ just updated) and $s_j = s_k$, two cases are distinguished.

- *Negative loop*: $b_j > b_k$. **V** loses the evaluation game.
- *Zero loop or a 2nd positive loop*: ($b_j = b_k$ & $l^+ = 0$) or ($b_j \leq b_k$ & $l^+ = 1$). **V** wins the evaluation game if $T \models \Psi$, and else $\overline{\mathbf{V}}$ wins.

In (ii) and (iii) the intuitive idea is that if one player can make a certain type of loop (without changing T), then such loops could be repeated infinitely - because otherwise the opponent should have been able to prevent the first such loop. Thus in (ii) the verifier is given unlimited budget, because (s)he has shown that arbitrarily many positive loops could be made. The rules in (iii) follow a similar idea.

(iv) Transition phase. First, $\mathbf{V}$ chooses a joint action $g \in D_A(s_k)$. Then, $\overline{\mathbf{V}}$ chooses a joint action $\overline{g} \in D_{\overline{A}}(s_k)$. Let $s_{k+1} := \delta(s_k, g \oplus \overline{g})$ and $b_{k+1} := b_k + wf(s_k, g \oplus \overline{g})$. If $b_{k+1} < 0$ and $l^+ = 0$, the game ends and $\mathbf{V}$ loses. Else, π is extended with the new pair (s_{k+1}, b_{k+1}).

This concludes the rules of the transition game. There are no infinite plays and each play ends as soon as at most two loops are detected. One can show that the length of the plays is bounded by $|S| \times (tw(\Psi) + 2)$. That is why, the transition game $\mathcal{TG}_{\text{LM}}(\mathbf{V}, s, A, b, \Psi)$ is said to be *loop-minimal* (LM). To be exact, the complete positions of the transition game should also include the current phase (i)–(iv) (and their subphases) for indicating how the rules are applied. However, we omitted this additional information for ease of presentation.

3.2 Capped Games

In this section, we define the resource-capped evaluation game $\mathcal{G}_{\text{RC}}[\beta](M, s, \phi)$ for some $\beta \in \mathbb{N}$ that differs from $\mathcal{G}_{\text{LM}}(M, s, \phi)$ only by its transition game. In the game $\mathcal{G}_{\text{RC}}[\beta](M, s, \phi)$, the transition game $\mathcal{TG}_{\text{RC}}[\beta](\mathbf{V}, s, A, b, \Psi)$ is parameterised by a *resource cap* $\beta \in \mathbb{N}$, which provides an upper bound on the resource value. When a transition game has a resource cap β, then we say that it is *β-capped*. The differences between $\mathcal{TG}_{\text{RC}}[\beta](\mathbf{V}, s, A, b, \Psi)$ and $\mathcal{TG}_{\text{LM}}(\mathbf{V}, s, A, b, \Psi)$ are defined below.

- Instead of having the state-resource history π in positions, there is only one pair (s', b'), containing the current state $s' \in S$ and the current resource value $b' \in \mathbb{Z}$. The initial pair is $(s, \min(b, \beta))$. The positions of the game are of the form $((s', b'), T')$ with initial truth function T_{open}.
- The verification phase (i) from $\mathcal{TG}_{\text{LM}}(\mathbf{V}, s, A, b, \Psi)$ is also included in the game $\mathcal{TG}_{\text{RC}}[\beta](\mathbf{V}, s, A, b, \Psi)$.
- There is no loop check phase in the game, i.e. the phases (ii) and (iii) from $\mathcal{TG}_{\text{LM}}(\mathbf{V}, s, A, b, \Psi)$ are not present in $\mathcal{TG}_{\text{RC}}[\beta](\mathbf{V}, s, A, b, \Psi)$ (no l^+ bit either).
- $\mathcal{TG}_{\text{RC}}[\beta](\mathbf{V}, s, A, b, \Psi)$ has also a transition phase (iv): **(iv) Transition phase.** $\mathbf{V}$ chooses a joint action $g \in D_A(s')$. Then $\overline{\mathbf{V}}$ chooses a joint action $\overline{g} \in D_{\overline{A}}(s')$. Let $s' := \delta(s', g \oplus \overline{g})$ and $b' := \min(\beta, b' + wf(s, g \oplus \overline{g}))$. If $b' < 0$, the game ends and $\mathbf{V}$ loses. Else, the play continues with the new pair (s', b') in the verification phase.

If a play of a transition game $\mathcal{TG}_{\text{RC}}[\beta](\mathbf{V}, s, A, b, \Psi)$ is infinite, then from some position onwards all positions must have the same truth function T (as truth functions can be updated only finitely often). $\mathbf{V}$ wins such a play iff $T \models \Psi$.

This version of the transition game is somewhat simpler because there is no need to keep track of state-resource histories or to check if any loop is made. However, these games last typically much longer because the players may do a huge numbers of loops (for accumulating/exhausting resources or just for prolonging the game out of spite). Besides, the update $b' := \min(\beta, b' + wf(s, g \oplus \overline{g}))$ is reminiscent to the way the accumulated weight is computed in the *lower-weak-upper-bound problem* (Bouyer et al. 2008; Hélouët, Markey, and Raha 2019). $\mathcal{TG}_{\text{RC}}[\beta](\mathbf{V}, s, A, b, \Psi)$ is said to be *resource-capped* (RC).

In $\mathcal{G}_{\text{RC}}[\beta](M, s, \phi)$, the number of positions of the form $(\mathbf{V}, r, \psi)$ is in $\mathcal{O}(|S| \cdot subf(\phi))$; the number of positions of the form $((r, b), T)$ is in $\mathcal{O}(subf(\phi) \cdot |S| \cdot \beta \cdot 3^{tw(\phi)})$. Hence, the number of positions of $\mathcal{G}_{\text{RC}}[\beta](M, s, \phi)$ is in $\mathcal{O}(|S| \cdot \beta \cdot subf(\phi) \cdot 3^{tw(\phi)})$. As truth functions in $\mathcal{G}_{\text{RC}}[\beta](M, s, \phi)$ are updated monotonously, we get the following.

Proposition 1. *(I) $\mathcal{G}_{RC}[\beta](M, s, \phi)$ can be reduced to Büchi games. (II) Given $K \in \mathbb{N}$ and a polynomial $P(\cdot)$, $\mathcal{G}_{RC}[\beta](M, s, \phi)$ can be solved in polynomial-time when $\beta \leq P(|M| + |\phi|)$ and $tw(\phi) \leq K$.*

To prove Proposition 1(I), we simply take the set of positions of the evaluation game (including transition games) as the set positions of the Büchi game and add a self-loop for all the winning positions. Player 1 (resp. Player 2) takes a turn where $\mathbf{A}$ (resp. $\mathbf{E}$) would take a turn. The winning set for Player 2 (which needs to be visited infinitely often) is defined as the union of: (a) ending positions where $\mathbf{E}$ wins; and (2) positions $((s, b), T)$ from the subgame $\mathcal{TG}_{\text{RC}}[\beta](\mathbf{V}, s, A, b, \Psi)$ such that $T \models \Psi$. To prove Proposition 1(II) it is sufficient to observe that Büchi games can be solved in polynomial-time when $\beta \leq P(|M| + |\phi|)$ and $tw(\phi) \leq K$ for some fixed K and $P(\cdot)$, $\mathcal{O}(|S| \times \beta \times subf(\phi) \times 3^{tw(\phi)})$ is polynomial in $|S| + subf(\phi)$.

3.3 Equivalence With Compositional Semantics

Let M be a CGS and ϕ be an $\mathsf{ATL}^+(1)$ formula. By $||M||^-_\infty$ we mean the *highest negative weight for a transition*, i.e. $||M||^-_\infty = \max\{|wf(s, f)| : s \in S,\ f \in D_{Ag}(s), wf(s, f) \leq 0\} \cup \{0\}$. We write $F(M, \phi)$ to denote $||M||^-_\infty(||M||^-_\infty(|S| \times tw(\phi))^2 + |S| \times tw(\phi))$. The correctness of $\mathcal{G}_{\text{LM}}(M, s, \phi)$ and of $\mathcal{G}_{\text{RC}}[\beta](M, s, \phi)$ for β large enough is one of our main results.

Theorem 2. **(I)** *$M, s \models \phi$ is equivalent to* **(II)** *and* **(III)**.
(II) ***E*** *has a winning strategy for $\mathcal{G}_{LM}(M, s, \phi)$.*
(III) ***E*** *has a winning strategy for $\mathcal{G}_{RC}[F(M, \phi)](M, s, \phi)$.*

$F(M, \phi)$ depends on $||M||^-_\infty$ but not on $||M||_\infty$ as what matters is the maximal decrement, similarly to the complexity analysis for the covering problem for VASS in (Rackoff 1978, Lemma 3) and (Demri et al. 2013, Lemma 4).

The equivalence between (I) and (II) is proven by structural induction on ϕ. We only need to consider the case $\phi = \langle\langle A^b \rangle\rangle \Psi$ since the other cases are proven exactly as in (Goranko, Kuusisto, and Rönnholm 2021). By way of example, if there is σ witnessing $M, s \models \langle\langle A^b \rangle\rangle \Psi$, we use forthcoming Lemma 5 to eliminate all unnecessary loops from σ, leading to a small strategy skeleton (see Section 4). Then, we show that a winning strategy for $\mathbf{E}$ in $\mathcal{TG}_{\text{LM}}(\mathbf{E}, s, A, b, \Psi)$ can be read from the small strategy skeleton. Similarly, to prove (I) equivalent to (III), we take advantage of forthcoming Lemma 7, which allows us to put in correspondence winning strategies witnessing $M, s \models \langle\langle A^b \rangle\rangle \Psi$ but truncated to $F(M, \phi)$ and strategies for $\mathbf{E}$ to win $\mathcal{TG}_{\text{RC}}[F(M, \phi)](\mathbf{E}, s, A, b, \Psi)$.

4 Representing Strategies Finitely

In this section, we provide developments about the compositional semantics, interesting for itself, but above all useful to establish the equivalence with evaluation games (see Theorem 2). It culminates with Lemma 6 that states an equivalence between satisfaction of $M^\star, s^\star \models \langle\langle A^{\star,b} \rangle\rangle \Psi^\star$ and the existence of a finite tree of polynomial depth and branching width (forthcoming tree $\mathbf{t}^\star_\sigma$), see a refinement in Lemma 7.

Let $M^\star$ be a CGS, $s^\star \in S$, $\langle\langle A^{\star,b} \rangle\rangle \Psi^\star$ be an $\mathsf{ATL}^+(1)$ state formula and σ be a strategy for $A^\star$. The set of computations starting at $s^\star$ and respecting σ can be organised as an infinite tree $\mathbf{t}_\sigma$. Such a tree is usually called a *strategy tree* (it summarises the choices made by the strategy). Our intention below is to define a *finite subtree* $\mathbf{t}^\star_\sigma$ of $\mathbf{t}_\sigma$ via the intermediate tree $\mathbf{t}^\omega_\sigma$. Assuming that $D^\perp_{Ag} \stackrel{\text{def}}{=} \bigcup_{s \in S} D_{Ag}(s) \uplus \{\perp\}$, a node in $\mathbf{t}_\sigma$ labelled by an element in $S \times \mathbb{Z} \times D^\perp_{Ag}$ records the last state of the history, the accumulated weight and the joint action leading to this state (joint actions are useful to compute weights). $\mathbf{t}_\sigma$ is defined as the smallest labelled tree $\mathbf{t}_\sigma : dom(\mathbf{t}_\sigma) \to S \times \mathbb{Z} \times D^\perp_{Ag}$ as follows (as usual $dom(\mathbf{t}_\sigma)$ is a prefix-closed subset of $\mathbb{N}^*$).

- $\varepsilon \in dom(\mathbf{t}_\sigma)$ and $\mathbf{t}_\sigma(\varepsilon) \stackrel{\text{def}}{=} (s_0, b, \perp)$ with $s_0 = s^\star$. $\perp$ is a dummy value used only at the root node.
- Assuming that the transitions of the form $s^\star \xrightarrow{\sigma(s^\star) \oplus g} r$ are $s^\star \xrightarrow{\sigma(s^\star) \oplus g_0} r_0, \ldots, s^\star \xrightarrow{\sigma(s^\star) \oplus g_{\alpha-1}} r_{\alpha-1}$ for some $\alpha \geq 1$, we require $0, \ldots, \alpha - 1 \in dom(\mathbf{t}_\sigma)$. For all $i \in \{0, \ldots, \alpha - 1\}$, $\mathbf{t}_\sigma(i) \stackrel{\text{def}}{=} (r_i, wf(s^\star, \sigma(s^\star) \oplus g_i) + b, \sigma(s^\star) \oplus g_i)$. Observe that $\alpha \leq \Delta(M^\star) \leq |M^\star|$.
- For the general case, assume that $\mathbf{n} \in dom(\mathbf{t}_\sigma)$, the label of the branch leading to $\mathbf{n}$ is $(s_0, n_0, f_0) \cdots (s_k, n_k, f_k)$ and the transitions of the form $s_k \xrightarrow{\sigma(s_0 \cdots s_k) \oplus g} r$ are $s_k \xrightarrow{\sigma(s_0 \cdots s_k) \oplus g_0} r_0, \ldots, s_k \xrightarrow{\sigma(s_0 \cdots s_k) \oplus g_{\alpha-1}} r_{\alpha-1}$. We require $\mathbf{n} \cdot 0, \ldots, \mathbf{n} \cdot (\alpha - 1) \in dom(\mathbf{t}_\sigma)$ and for all $i \in \{0, \ldots, \alpha - 1\}$, $\mathbf{t}_\sigma(\mathbf{n} \cdot i)$ is equal to
$$(r_i, wf(s_k, \sigma(s_0 \cdots s_k) \oplus g_i) + n_k, \sigma(s_0 \cdots s_k) \oplus g_i).$$

The *S-label* of the sequence $(s_0, n_0, f_0) \cdots (s_k, n_k, f_k) \ldots$ is the sequence $s_0 s_1 \cdots s_k \cdots$. Lemma 3 is by an easy verification and summarises the main properties of $\mathbf{t}_\sigma$.

Lemma 3. *(I) For all $\lambda \in comp(s^\star, \sigma)$, there is an infinite branch in $\mathbf{t}_\sigma$ whose S-label is λ. (II) For every infinite branch of $\mathbf{t}_\sigma$, its S-label is a computation in $comp(s^\star, \sigma)$. (III) The following are equivalent: (1) for all the computations $\lambda \in comp(s^\star, \sigma)$, $M^\star, \lambda \models \Psi^\star$; (2) for every infinite branch of $\mathbf{t}_\sigma$ with S-label π, $M^\star, \pi \models \Psi^\star$.*

Below, we define the functions U_0, U_1 and $U_{>1}$ that update *deterministically* a truth function depending on which state formulae from $atoms(\Psi^\star)$ hold true on a state (corresponding to the verification phase in transition games with systematic and correct updates). This is essential to define next the trees $\mathbf{t}^\omega_\sigma$. The strategy trees $\mathbf{t}_\sigma$ shall be enriched with truth functions on nodes, while updating the truth functions using U_0, U_1 or $U_{>1}$. We need to distinguish three functions to handle maximal state formulae in $atoms(\Psi^\star)$ not in the scope of X and U, from the X-formulae, and finally from the Boolean combinations of U-formulae. Let $s \in S$ be a state and T be a truth function. We write $U_0(s, T)$ to denote the truth function obtained from T according to clauses below.

($\mathcal{C}_{st}$) If $\phi \in atoms(\Psi^\star)$, $T(\phi) = open$ and $M^\star, s \models \phi$ (resp. $M^\star, s \not\models \phi$), then $U_0(s, T)(\phi) = \top$ (resp. $U_0(s, T)(\phi) = \perp$).

($\mathcal{C}_{\mathsf{U}}$) If $\phi_1 \mathsf{U} \phi_2 \in atoms(\Psi^\star)$, $T(\phi_1 \mathsf{U} \phi_2) = open$ and $M^\star, s \models \phi_2$ (resp. $M^\star, s \not\models \phi_2 \vee \phi_1$), then $U_0(s, T)(\phi_1 \mathsf{U} \phi_2) = \top$ (resp. $U_0(s, T)(\phi_1 \mathsf{U} \phi_2) = \perp$).

($\mathcal{C}_{oth}$) For other temporal atoms, $U_0(s, T)$ is defined as T.

We introduce $U_1(s, T)$ as a slight variant of $U_0(s, T)$ handling X-formulae. We write $U_1(s, T)$ to denote the truth function obtained from T according to the clauses ($\mathcal{C}_{st}$) and ($\mathcal{C}_{\mathsf{U}}$) and the new clause ($\mathcal{C}_{\mathsf{X}}$).

($\mathcal{C}_{\mathsf{X}}$) If $\mathsf{X}\, \phi_1 \in atoms(\Psi^\star)$, $T(\mathsf{X}\, \phi_1) = open$ and $M^\star, s \models \phi_1$ (resp. $M^\star, s \not\models \phi_1$), then $U_1(s, T)(\mathsf{X}\, \phi_1) = \top$ (resp. $U_1(s, T)(\mathsf{X}\, \phi_1) = \perp$).

We write $U_{>1}(s, T)$ to denote the function based on ($\mathcal{C}_{\mathsf{U}}$) and ($\mathcal{C}_{oth}$) (assuming T is already total over the set of X-formulae and state formulae). We write $U(\lambda, T)$ to denote the repeated effect of updating T via $\lambda \in S^+$.

- If λ has length 1, then $U(\lambda, T) = U_0(\lambda(0), T)$.
- If λ has length 2, then $U(\lambda, T) = U_1(\lambda(1), U_0(\lambda(0), T))$.
- Otherwise, $U(\lambda, T) = U_{>1}(\lambda(n), U(\lambda[0, n-1], T))$.

We write $S(T)$ to denote the set $\{U(\lambda, T) \mid \lambda \in S^+\}$. The rationale behind its definition is best explained below.

Lemma 4. *Let λ be an infinite computation of $M^\star$ from $s^\star$. $M^\star, \lambda \models \Psi^\star$ iff there is $I \geq 0$ such that for all $J \geq I$, we have $U(\lambda[0, J], T_{open}) = U(\lambda[0, I], T_{open})$ and $U(\lambda[0, I], T_{open}) \models \Psi^\star$.*

Let σ be a strategy for $A^\star$, $s^\star \in S$ and $b \in \mathbb{N}$. Below, we define the tree $\mathbf{t}^\omega_\sigma : dom(\mathbf{t}^\omega_\sigma) \to S \times S(T_{open}) \times (\mathbb{Z} \cup \{\omega\}) \times D^\perp_{Ag}$ as an enriched version of $\mathbf{t}_\sigma$ in which each label contains a truth function and we also introduce the value ω that stands for "unbounded". The symbol ω has a role similar to the value ω in Karp-Miller trees (Karp and Miller 1969) dedicated to Petri nets. Below, we construct the infinite tree $\mathbf{t}^\omega_\sigma$ possibly admitting the resource value ω (with $\omega = n + \omega = \omega + n$ for all $n \in \mathbb{Z}$) and then we define a *finite tree* $\mathbf{t}^\star_\sigma$ from $\mathbf{t}^\omega_\sigma$ by truncating its branches. Typically, if we are about to add a new node $\mathbf{n}$ on a branch with label (r, T, n, f) and there is an ancestor node with label (r, T, n', f') and $n' < n$ (cycle with positive accumulated weight), then the label of $\mathbf{n}$ is actually set to (r, T, ω, f). By construction, all the nodes below this new node has resource value ω. We explain below how to build $\mathbf{t}^\omega_\sigma$.

- $\varepsilon \in dom(\mathbf{t}^\omega_\sigma)$ and $\mathbf{t}^\omega_\sigma(\varepsilon) \stackrel{\text{def}}{=} (s_0, T_0, b, \perp)$ with $s_0 = s^\star$ and $T_0 = U_0(s^\star, T_{open})$.
- Assuming that the transitions of the form $s^\star \xrightarrow{\sigma(s^\star) \oplus g} r$ are $s^\star \xrightarrow{\sigma(s^\star) \oplus g_0} r_0, \ldots, s^\star \xrightarrow{\sigma(s^\star) \oplus g_{\alpha-1}} r_{\alpha-1}$, we have $0, \ldots, \alpha - 1 \in dom(\mathbf{t}^\omega_\sigma)$. For all $i \in \{0, \ldots, \alpha - 1\}$ s.t.

$$\overbrace{(r_i, U_1(r_i, U_0(s^\star, T_{open}))) \neq (s^\star, U_0(s^\star, T_{open}))}^{\text{progress in the truth function}} \text{ or } \underbrace{wf(s^\star, \sigma(s^\star) \oplus g_i) \leq 0}_{\text{non-positive weight}},$$

$\mathbf{t}^\omega_\sigma(i) \stackrel{\text{def}}{=} (r_i, T^1_i, wf(s^\star, f^1_i) + b, f^1_i)$ with $T^1_i = U_1(r_i, U_0(s^\star, T_{open}))$ and $f^1_i = \sigma(s^\star) \oplus g_i$. If $(r_i, T^1_i) = (s^\star, U_0(s^\star, T_{open}))$ and $wf(s^\star, f^1_i) > 0$, $\mathbf{t}^\omega_\sigma(i) \stackrel{\text{def}}{=} (r_i, T^1_i, \omega, f^1_i)$. ω is introduced if a loop with positive weight is detected with no progress on the truth function.

- For the general case, assume that $\mathbf{n} \in dom(\mathbf{t}^\omega_\sigma)$, the label of the finite branch leading to $\mathbf{n}$ is $(s_0, T_0, n_0, f_0) \cdots (s_k, T_k, n_k, f_k)$ (n_k may be equal to ω). Moreover, assuming that the transitions of the form $s_k \xrightarrow{\sigma(s_0 \cdots s_k) \oplus g} r$ are $s_k \xrightarrow{\sigma(s_0 \cdots s_k) \oplus g_0} r_0$, ..., $s_k \xrightarrow{\sigma(s_0 \cdots s_k) \oplus g_{\alpha-1}} r_{\alpha-1}$, for some $\alpha \geq 1$, then $\mathbf{n} \cdot 0, \ldots, \mathbf{n} \cdot (\alpha - 1) \in dom(\mathbf{t}^\omega_\sigma)$. For all $i \in \{0, \ldots, \alpha - 1\}$ such that $n_k = \omega$ or *there is no* $j < k$ such that $(s_j, T_j) = (r_i, U_{>1}(r_{i+1}, T_k))$ and $n_j < wf(s_k, \sigma(s_0 \cdots s_k) + g_i) + n_k$, $\mathbf{t}^\omega_\sigma(\mathbf{n} \cdot i) \stackrel{\text{def}}{=} (r_i, U_{>1}(r_i, T_k), wf(s_k, \sigma(s_0 \cdots s_k) \oplus g_i) + n_k, \sigma(s_0 \cdots s_k) \oplus g_i)$. This case occurs when no ω is introduced. Otherwise, when there is some $j < k$ satisfying such conditions and $n_k \neq \omega$,

$$\mathbf{t}^\omega_\sigma(\mathbf{n} \cdot i) \stackrel{\text{def}}{=} (r_i, U_{>1}(r_i, T_k), \omega, \sigma(s_0 \cdots s_k) \oplus g_i).$$

Let us mention a few properties about $\mathbf{t}^\omega_\sigma$: for every branch of $\mathbf{t}^\omega_\sigma$ with label $(\cdot, T_0, \cdot) \cdot (\cdot, T_1, \cdot) \cdots$, there is $I \geq 0$ such that $T_I = T_{I+1} = \cdots$, $|\{T_i \mid i \geq 0\}| \leq tw(\Psi^\star)$ and the branching width of $\mathbf{t}^\omega_\sigma$ is bounded by $\Delta(M^\star)$.

Below, we define $\mathbf{t}^\star_\sigma$ as a subtree of $\mathbf{t}^\omega_\sigma$. Intuitively, $\mathbf{t}^\star_\sigma$ is obtained from $\mathbf{t}^\omega_\sigma$ by truncating every branch if no more progress is to be expected on the branch and enough pieces of information is preserved to repeat the final part of the truncated finite branch. On each infinite branch of $\mathbf{t}^\omega_\sigma$, we identify a unique terminating node from which the strict subtree is truncated. $\mathbf{t}^\star_\sigma$ could be designed directly from σ without defining $\mathbf{t}^\omega_\sigma$ but we feel it is easier that way to grasp the whole construction. As $\mathbf{t}^\omega_\sigma$ is finite-branching, by König's Lemma, this allows us to guarantee that $\mathbf{t}^\star_\sigma$ is a finite labelled tree. If $\mathbf{t}^\omega_\sigma$ contains a negative value for some $\mathbf{n} \in dom(\mathbf{t}^\omega_\sigma)$, then $\mathbf{t}^\star_\sigma$ is defined as a single root node with label equal to $\mathbf{t}^\omega_\sigma(\varepsilon)$ (dummy value). Otherwise, we assume that all weights in $\mathbf{t}^\omega_\sigma$ are non-negative and let $\mathcal{B} = i_1 i_2 i_3 \cdots$ be a branch of $\mathbf{t}^\omega_\sigma$ with label $(r_0, T_0, n_0, \perp) \cdot (r_1, T_1, n_1, f_1) \cdots$. We make a case analysis to identify the terminating node on $\mathcal{B}$. There is always a minimal position I_{stab} such that for all $K \geq I_{stab}$, we have $T_K = T_{I_{stab}}$.

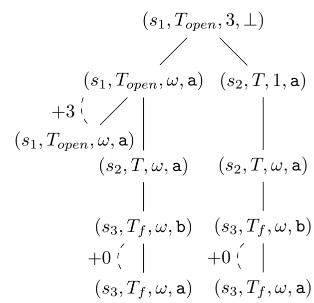
Case $n_i = \omega$ **for some** i. There is a minimal position $I^\omega_{stab} \geq I_{stab}$ such that for all $K \geq I^\omega_{stab}$, we have $n_K = n_{I^\omega_{stab}} = \omega$ and $T_K = T_{I^\omega_{stab}}$. We distinguish two cases. If there are no $I^\omega_{stab} \leq K < K'$ such that $(r_K, T_K) = (r_{K'}, T_{K'})$ and $\Sigma^{K'-1}_{\ell=K} wf(r_\ell, f_{\ell+1}) \geq 0$, then J is the first position strictly after I^ω_{stab} for which there is $I^\omega_{stab} \leq J' < J$ such that $(r_{J'}, T_{J'}) = (r_J, T_J)$. Such a position exists by the Pigeonhole Principle.

Otherwise, J is the first position strictly after I^ω_{stab} for which there exists $I^\omega_{stab} \leq J' < J$ such that $(r_{J'}, T_{J'}) = (r_J, T_J)$ and $\Sigma^{J-1}_{\ell=J'} wf(r_\ell, f_{\ell+1}) \geq 0$.

Case $n_i \neq \omega$ **for all** i. Since ω is not in $\mathcal{B}$, no cycle with positive weight occurs on $\mathcal{B}$. Hence, for all $K' < K$ such that $(r_K, T_K) = (r_{K'}, T_{K'})$, $n_K \leq n_{K'}$. Since no node along the branch is labelled by a negative value in $\mathbf{t}^\omega_\sigma$, for all $(r, T) \in S \times S(T_{open})$, there is a minimal position $I_{(r,T)}$ after the position I_{stab} such that for all $I_{(r,T)} \leq K < K'$ with $(r_K, T_K) = (r_{K'}, T_{K'}) = (r, T)$, we have $n_K = n_{K'}$ (again by the Pigeonhole Principle). Let J be the first position after I_{stab} for which there is $I_{stab} \leq J' < J$ such that $(r_{J'}, T_{J'}, n_{J'}) = (r_J, T_J, n_J)$ $(J - J' \leq |S|)$.

The *terminating* node of $\mathcal{B}$ is $\mathbf{n}_J = i_1 i_2 i_3 \cdots i_J$.

Below, we present $\mathbf{t}^\star_\sigma$ for the CGS in Section 2 for checking $M, s_1 \models \langle\!\langle \{1\}^3 \rangle\!\rangle (\mathsf{G}\, p_1 \vee \mathsf{F}\, p_2)$ with the (non-winning) strategy σ for the agent 1 that chooses the action $\mathtt{a}$ on s_2 only if no state is visited twice, otherwise $\mathtt{b}$.



T denotes $[\top \mathsf{U} \neg p_1 \mapsto \top, \top \mathsf{U}\, p_2 \mapsto open]$ and T_f denotes the map $[\top \mathsf{U} \neg p_1 \mapsto \top, \top \mathsf{U}\, p_2 \mapsto \top]$. Joint actions for 1 are represented by the action choosen by 1. E.g., $T_f \models (\mathsf{G}\, p_1 \vee \mathsf{F}\, p_2)$ and $T \not\models (\mathsf{G}\, p_1 \vee \mathsf{F}\, p_2)$.

In all cases, $T_{J'} = \cdots = T_J$ and $r_{J'} = r_J$. These observations can be organised a bit more. Below, we introduce several potential properties about branches of $\mathbf{t}^\star_\sigma$ labelled by $(s_0, T_0, n_0, \perp) \cdot (s_1, T_1, n_1, f_1) \cdots (s_K, T_K, n_K, f_K)$.

(a) $\{n_0, n_1, n_2, \ldots, n_K\} \subseteq \mathbb{N} \cup \{\omega\}$. (no negative value)

(b) There is $I < K$ such that $T_I = T_K$, $T_I \models \Psi^\star$, $(s_I, T_I, n_I) = (s_K, T_K, n_K)$ and $\sum_{j=I+1}^{K} wf(s_{j-1}, f_j) \geq 0$. (satisfaction of $\Psi^\star$ guaranteed and non-negative final cycle)

(c) There are no $0 \leq J' < J < K$ such that $(s_{J'}, T_{J'}) = (s_J, T_J)$ and $\sum_{j=J'+1}^{J} wf(s_{j-1}, f_j) \leq 0$. (No pair (r, T) visited twice without progress on the accumulated weight)

(d) There are no $0 \leq J' < J < K$ such that $(s_{J'}, T_{J'}, n_{J'}) = (s_J, T_J, n_J)$ and $\sum_{j=J'+1}^{J} wf(s_{j-1}, f_j) > 0$. (no positive internal cycle containing only ω)

(e) $K \leq |S| \times (tw(\Psi^\star) + 2)$. (small depth)

(f) $K - I \leq |S|$ and (b). (small final cycle + (b))

If σ is a strategy witnessing the satisfaction of $M^\star, s^\star \models \langle\!\langle A^{\star,b} \rangle\!\rangle \Psi^\star$, one can show that $\mathbf{t}^\star_\sigma$ satisfies (a)–(b). Observe that $\mathbf{t}^\star_\sigma$ defined earlier for the CGS in Section 2 satisfies (a)–(f) but for the strategy σ' in which the agent 1 chooses twice the action $\mathtt{a}$ on s_2 and then the action $\mathtt{b}$ the third time, $\mathbf{t}^\star_{\sigma'}$ *violates (d) though* σ' *witnesses the satisfaction* of $M, s_1 \models \langle\!\langle \{1\}^3 \rangle\!\rangle (\mathsf{G}\, p_1 \vee \mathsf{F}\, p_2)$. In a way, σ' made an unnecessary detour by staying on s_2 too long. But more importantly, one can always find a strategy satisfying all the conditions (a)–(f), which is computationally appealing because $\mathbf{t}^\star_\sigma$ is *small* (see (e), (f) and branching width bounded by $|M^\star|$).

Lemma 5. *If $M^\star, s^\star \models \langle\langle A^{\star,b}\rangle\rangle\Psi^\star$, then there is a strategy σ' such that all the branches of $\mathbf{t}^\star_{\sigma'}$ satisfy (a)–(f).*

If $M^\star, s^\star \models \langle\langle A^{\star,b}\rangle\rangle\Psi^\star$ is witnessed by σ, then $\mathbf{t}^\star_{\sigma'}$ in Lemma 5 is called a *small strategy skeleton*. Its proof consists in eliminating the negative cycles in $\mathbf{t}^\star_\sigma$ (on the alphabet $S \times S(T_{open})$), which requires some care because there might be an infinite amount, then we eliminate a finite amount of other cycles, see condition (d). Now, we state the key result to get the PSPACE upper bound for MC($\mathsf{ATL}^+(1)$) and to show equivalence with the game-theoretical semantics. Indeed, (a)–(f) can be put in correspondence with winning conditions for **E** in loop-minimal transition games.

Lemma 6. *(I) $M^\star, s^\star \models \langle\langle A^{\star,b}\rangle\rangle\Psi^\star$ iff (II) there is a strategy σ such that all the branches of $\mathbf{t}^\star_\sigma$ satisfy (a)–(f).*

The proof of "(II) $\Rightarrow$ (I)" eliminates ω from $\mathbf{t}^\star_\sigma$ to design a strategy σ_0 witnessing the satisfaction of $M^\star, s^\star \models \langle\langle A^{\star,b}\rangle\rangle\Psi^\star$. First, a finite tree $\mathbf{t}^{\star\star}$ is built from $\mathbf{t}^\star_\sigma$ (but with precise resource values), its depth bounded by $||M||^-_\infty \times (|S| \times tw(\Psi^\star))^2 + (|S| \times tw(\Psi^\star))$. Then, σ_0 is defined from $\mathbf{t}^{\star\star}$ by imposing regularity to the strategy.

By Lemma 6, determining whether $M^\star, s^\star \models \langle\langle A^{\star,b}\rangle\rangle\Psi^\star$ holds, amounts to guess a tree of branching width at most $\Delta(M^\star)$ and of depth polynomial in the size of $\Psi^\star$ and in $|S|$. The conditions about cycles require to keep in memory a branch of polynomial length. This can be computed with an alternating Turing machine running in polynomial-time, leading to PSPACE (assuming the satisfaction of maximal state formulae is known). Interestingly, Lemma 6 provides an original viewpoint to decide $M^\star, s^\star \models \langle\langle A^\star\rangle\rangle\Psi^\star$ for ATL^+ (no more initial budget b). Indeed, assuming that $\mathbf{t}^\star_\sigma$ for ATL^+ is designed without ω and the nodes are only labelled by pairs (s, T) (no resource value, no joint action to compute accumulated weights), the restriction of Lemma 6 to ATL^+ becomes: $M^\star, s^\star \models \langle\langle A^\star\rangle\rangle\Psi^\star$ iff there is σ such that for every maximal branch of $\mathbf{t}^\star_\sigma$ labelled by $(s_0, T_0)\cdots(s_K, T_K)$, $K \leq |S| \times tw(\Psi^\star)$, there are no $0 \leq J < J' < K$ such that $(s_J, T_J) = (s_{J'}, T_{J'})$ and there is $I < K$ such that $T_I = T_K$ and $T_I \models \Psi^\star$. The *only possibility* to see a pair (s, T) twice on a branch is to occur the second time on the leaf with $T \models \Psi^\star$. This simple criterion leads to PSPACE for MC(ATL^+) by using a standard labelling algorithm. This contrasts with the sophisticated evaluation games in (Goranko, Kuusisto, and Rönnholm 2021) and proof system in (Cerrito 2021) to handle MC(ATL^+).

Below, we refine the developments about $\mathbf{t}^\omega_\sigma$ by truncating the resource values above a fixed resource value β and giving up the use of ω, possibly at the cost of introducing negative values because previous values along a branch are truncated. On the positive side, we show that for a value β polynomial in $||M||^-_\infty$, $|S|$ and $tw(\Psi^\star)$, $M^\star, s^\star \models \langle\langle A^{\star,b}\rangle\rangle\Psi^\star$ holds iff there is σ such that $\mathbf{t}^\omega_\sigma$ truncated to β has no negative values and for all branches, the final truth function satisfies $\Psi^\star$. Since the set of possible labels in such a truncated tree is finite and equal to $S \times S(T_{open}) \times [0, \beta] \times D^\perp_{Ag}$, the satisfaction of $M^\star, s^\star \models \langle\langle A^{\star,b}\rangle\rangle\Psi^\star$ is equivalent to the existence of a strategy in a Büchi game, whose accepting locations are labelled by truth functions T satisfying $\Psi^\star$. More importantly, if the cardinality of $S \times S(T_{open}) \times [0, \beta] \times D^\perp_{Ag}$ is polynomial (typically by bounding β and $|S(T_{open})|$, $|S|$ and $|D^\perp_{Ag}|$ being linear in $|M^\star|$), then we can identify subproblems of MC($\mathsf{ATL}^+(1)$) in PTIME because Büchi games are in PTIME (Chatterjee, Henzinger, and Piterman 2008).

Let $\beta \in \mathbb{N}$ and σ be a strategy for $A^\star$. We intend to determine when $M^\star, s^\star \models \langle\langle A^{\star,b}\rangle\rangle\Psi^\star$ holds true. We write $\mathbf{tt}^\beta_\sigma$ to denote the *truncated strategy tree* obtained from $\mathbf{t}^\omega_\sigma$ by giving up the use of ω and each new resource value of the form $wf(s_k, \sigma(s_0\cdots s_k)\oplus g)+n_k$ in $\mathbf{t}^\omega_\sigma$ is actually replaced by its β-truncation $\min(\beta, wf(s_k, \sigma(s_0\cdots s_k)\oplus g)+n_k)$. The updates for the control states, the truth functions and the joint actions are done in $\mathbf{tt}^\beta_\sigma$ as in $\mathbf{t}^\omega_\sigma$. For all $\mathbf{n} \in dom(\mathbf{t}^\omega_\sigma)$ with $\mathbf{t}^\omega_\sigma(\mathbf{n}) = (r, T, n, f)$ and $\mathbf{tt}^\beta_\sigma(\mathbf{n}) = (r, T, \overline{n}, f)$, we have $n \geq \overline{n}$. Even if $\mathbf{t}^\omega_\sigma$ does not contain negative values, the β-truncations may lead to negative values in $\mathbf{tt}^\beta_\sigma$.

Lemma 7 below refines Lemma 6 and its consequences include the equivalence between (I) and (III) in Theorem 2 and the design of PTIME fragments for MC($\mathsf{ATL}^+(1)$).

Lemma 7. *$M^\star, s^\star \models \langle\langle A^{\star,b}\rangle\rangle\Psi^\star$ iff there is a strategy σ for $A^\star$ such that $\mathbf{tt}^\beta_\sigma$ with $\beta = F(M^\star, \Psi^\star)$ has no negative values and all branches stabilise to some T satisfying $\Psi^\star$.*

$F(M^\star, \Psi^\star)$ is exponential in $|M^\star|$ and polynomial in $|\Psi^\star|$. We believe that a value β in $\mathcal{O}(||M^\star||^-_\infty \times |S| \times tw(\Psi^\star))$ could replace the one in Lemma 7 but our current proof of Lemma 7 uses more direct arguments, at the cost of having a non-optimal value for $F(M^\star, \Psi^\star)$.

5 From Model-Checking to Synthesis

In this section, we take advantage of Lemmas 6 and 7 and Theorem 2 to obtain new results about $\mathsf{ATL}^+(1)$.

5.1 On the Complexity of Model-Checking

The main complexity result of this work is stated below.

Theorem 8. *MC($\mathsf{ATL}^+(1)$) is PSPACE-complete.*

The PSPACE upper bound for MC($\mathsf{ATL}^+(1)$) can be established directly from Lemma 6 by using a standard labelling algorithm. The proof takes advantage of the equivalence between the compositional semantics and the evaluation games (whose proof is based on Lemma 6). Theorem 9 below takes advantage of Lemma 7 used in the proof for the equivalence between (I) and (III) in Theorem 2.

Theorem 9. *Given $K \in \mathbb{N}$ and a polynomial $P(\cdot)$, the model-checking problem is in PTIME for the fragment of $\mathsf{ATL}^+(1)$ s.t. $||M||^-_\infty \leq P(|M| + |\phi|)$ and $tw(\phi) \leq K$.*

If $||M||^-_\infty \leq P(|M| + |\phi|)$ and $tw(\phi) \leq K$ then the number of positions in $\mathcal{TG}_{\rm RC}[\beta_0](\mathbf{V}, s, A, b, \Psi)$ with $\beta_0 = F(M, \phi)$, $\langle\langle A^b\rangle\rangle\Psi \in subf(\phi)$ is polynomial in $|M| + |\phi|$. So, $\mathcal{G}_{\rm RC}[\beta_0](M, s, \phi)$ has also polynomial size in $|M| + |\phi|$. Since Büchi games can be solved in PTIME, the bound PTIME in Theorem 9 follows from Theorem 2 and Proposition 1. As a corollary, due to syntactic restrictions on $\mathsf{ATL}(1)$, we get the following.

Corollary 10. *MC($\mathsf{ATL}(1)$) is* PTIME*-complete, when restricted to the instances s.t.* $||M||_\infty^- \leq P(|M| + |\phi|)$ *for some fixed polynomial* $P(\cdot)$.

We argue that the assumption of polynomially large resource weights is natural for typical models and thus the complexity of this problem can go beyond PTIME only in the "pathological cases" where weights in the model are exponentially large. Our results can be adapted to variants of $\mathsf{ATL}^+(1)$, e.g. we can conclude MC($\mathsf{ATL}_p^+(1)$) in PSPACE.

5.2 Synthesis of Arithmetical Constraints

A standard way in formal verification to consider problems that go beyond model-checking, see e.g. (Bruyère and Raskin 2007), consists in replacing concrete numerical values by parameters and to synthesize constraints whose solutions solve positively instances of the model-checking problem. The synthesis of such constraints is not always possible but for MC($\mathsf{ATL}^+(1)$), we can take advantage of Lemma 7 to establish that there is $n \in \mathbb{N}$ such that $M, s \models \langle\!\langle A^n \rangle\!\rangle \Psi$ iff there is $n \leq F(M, \Psi)$ such that $M, s \models \langle\!\langle A^n \rangle\!\rangle \Psi$.

We introduce the logic $\mathsf{ParATL}^+(1)$, a parameterised version of $\mathsf{ATL}^+(1)$, following the approach from (Alechina et al. 2018, Section 6). Budget values b are replaced by variables among $\mathtt{x}_1, \ldots, \mathtt{x}_n, \ldots$. In a $\mathsf{ParATL}^+(1)$ formula, a variable $\mathtt{x}$ can occur in different places. We write $M, s \models \phi[\mathtt{x}_1 \leftarrow b_1, \ldots, \mathtt{x}_m \leftarrow b_m]$ to denote the satisfaction of ϕ from the parameterised version in which each $\mathtt{x}_i$ is replaced by b_i. Theorem 11 below states that we can characterise

$$\{\vec{b} \in \mathbb{N}^m \mid M, s \models \phi[\mathtt{x}_1 \leftarrow \vec{b}(1), \ldots, \mathtt{x}_m \leftarrow \vec{b}(m)]\}$$

with a Boolean combination of atomic constraints of the form $\mathtt{x}_i \geq B$ with $i \in [1, m]$. For example, the constraint on $\mathtt{x}$ to satisfy $M, s_1 \models \langle\!\langle \{1\}^{\mathtt{x}} \rangle\!\rangle (\mathsf{G}\, p_1 \vee \mathsf{F}\, p_2)$ with M defined in Section 2 is simply "$\mathtt{x} \geq 2$". Such a result helps to answer more questions: we can decide whether the set is infinite or whether it contains an element $\vec{b}$ satisfying a given Presburger formula $\psi(\mathtt{y}_1, \ldots, \mathtt{y}_m)$. For instance, suppose that $C(\mathtt{x}_1, \ldots, \mathtt{x}_m)$ is a constraint such that for all $\vec{b} \in \mathbb{N}^m$, we have $\vec{b} \models C$ iff $M, s \models \phi[\mathtt{x}_1 \leftarrow \vec{b}(1), \ldots, \mathtt{x}_m \leftarrow \vec{b}(m)]$. Checking the infinity of the set $\{\vec{b}(3) \in \mathbb{N} \mid M, s \models \phi[\mathtt{x}_1 \leftarrow \vec{b}(1), \ldots, \mathtt{x}_m \leftarrow \vec{b}(m)]\}$ can be done by checking the satisfaction of $\forall\, \mathtt{y}\; \exists\, \mathtt{z}_1, \ldots, \mathtt{z}_m\; (\mathtt{y} < \mathtt{z}_3) \wedge C(\mathtt{z}_1, \ldots, \mathtt{z}_m)$.

Theorem 11. *Let ϕ be a formula in $\mathsf{ParATL}^+(1)$ built over the variables $\mathtt{x}_1, \ldots, \mathtt{x}_m$, M be a CGS and $s \in S$. One can build in polynomial space in $|\phi| + |M|$ an arithmetical constraint $C(\mathtt{x}_1, \ldots, \mathtt{x}_m)$ built over atomic constraints of the form $\mathtt{x} \geq B$, such that for all $\vec{b} \in \mathbb{N}^m$, we have $\vec{b} \models C(\mathtt{x}_1, \ldots, \mathtt{x}_m)$ iff $M, s \models \phi[\mathtt{x}_1 \leftarrow \vec{b}(1), \ldots, \mathtt{x}_m \leftarrow \vec{b}(m)]$.*

5.3 Energy Game with Depth-One Objectives

In this section, we show that a simple energy game problem very closely related to $\mathsf{ATL}^+(1)$ can be solved in PSPACE. A *game graph* G is a structure (V, V_1, V_2, E, L) such that V is a finite set of vertices, $\{V_1, V_2\}$ is a partition of V, E is a finite set of edges from $V \times \mathbb{Z} \times V$ such that every vertex has at least one outgoing edge and $L : \mathrm{AP} \to \mathcal{P}(V)$.

Two players (1 and 2) move a token on G and a *configuration* of G is a pair (v, b) in $V \times \mathbb{Z}$. If the token is on a vertex v in $V_{\mathbf{P}}$, then player $\mathbf{P}$ chooses an edge (v, u, v') starting from v and move the token to v' and update the value accordingly. A *play* Π is an infinite sequence of configurations $(v_0, b_0), (v_1, b_1), \ldots$ such that $b_0 = 0$ and for all $i > 0$, we have $(v_{i-1}, b_i - b_{i-1}, v_i) \in E$. A *path* π is a non-empty finite prefix of a play, and we write $\Pi_{|n}$ to denote the path $(v_0, b_0), (v_1, b_1), \ldots, (v_n, b_n)$. Given a player $\mathbf{P} \in \{1, 2\}$, a strategy σ takes as input paths of the form $\pi \cdot (v, b)$ with $v \in V_{\mathbf{P}}$ and returns an edge in E of the form (v, u, v'). A play Π *respects* the strategy σ iff for all $i \in \mathbb{N}$ such that $v_i \in V_{\mathbf{P}}$, we have $\sigma(\Pi_{|i}) = (v_i, b_{i+1} - b_i, v_{i+1})$.

A *depth-one objective* Ψ is a Boolean combination of formulae of the form ϕ, $\mathsf{X}\, \phi$, and $\phi_1 \,\mathsf{U}\, \phi_2$ where the ϕ's are propositional formulae. We consider the satisfaction relation $\Pi \models \Psi$ with the infinite play Π following the LTL semantics. Given G, $v \in V$ and Ψ, *player 1 has a winning strategy* iff there is a strategy σ such that all the infinite plays Π starting from $(v, 0)$ and respecting σ, we have $\Pi \models \Psi$ *and* the counter values are never negative. Here is the *energy game problem with depth-one LTL objectives* (written $\mathcal{P}$): given $G = (V, V_1, V_2, E, L)$, $v \in V$ and a depth-one objective Ψ, is there a winning strategy for the player 1? $\mathcal{P}$ is a fragment of more general energy games, see e.g. (Colcombet et al. 2017) and here is a consequence of Theorem 8.

Theorem 12. *$\mathcal{P}$ is* PSPACE*-complete.*

PSPACE-hardness is due to graph games (without resources) with objectives $\bigwedge_i \top \,\mathsf{U}\, p_i$ (Alur and La Torre 2001, Theorem 4.4), PSPACE-membership is by reduction to MC($\mathsf{ATL}^+(1)$) (to instances $M, s \models \langle\!\langle A^0 \rangle\!\rangle \Psi$ with $msf(\Psi)$ made of propositional formulae). For objectives $\mathsf{G}\, p$ or $\top \,\mathsf{U}\, p$, the problem is known in NP $\cap$ co-NP (Chatterjee, Doyen, and Henzinger 2017) and positional strategies suffice for objectives $\mathsf{G}\, p$, see (Bouyer et al. 2008, Lemma 10).

6 Conclusion

We studied an extension of ATL^+ with one resource in which actions of the agents may produce or consume resources.

We introduced two evaluation games for $\mathsf{ATL}^+(1)$ formulae that are equivalent to the compositional semantics, see Theorem 2. This extends (and sometimes simplifies) what is done for ATL^+ in (Goranko, Kuusisto, and Rönnholm 2021) and the two transition games handle the resource values differently. The resource-capped game is particularly useful to design PTIME fragments of MC($\mathsf{ATL}^+(1)$) and we have shown that the game is equivalent to the compositional semantics whenever the resource cap β is above $F(M, \phi)$.

We have shown that MC($\mathsf{ATL}^+(1)$) is in PSPACE, and this is done by designing small strategy skeletons (see Lemma 6). As a by-product, it provides a simplified decision procedure for MC(ATL^+). Furthermore, we introduced a parameterised version $\mathsf{ParATL}^+(1)$ of $\mathsf{ATL}^+(1)$ for which the synthesis of Presburger formulae is done in polynomial-space (Theorem 11). Moreover, we briefly explained how the PSPACE bound for MC($\mathsf{ATL}^+(1)$) can be used to solve in polynomial-space an energy game problem with one counter and LTL objectives of temporal depth one (Theorem 12).

Acknowledgments

We thank the anonymous reviewers for the comments and suggestions that help us to improve the document.

The work of Raine Rönnholm was supported by Jenny and Antti Wihuri Foundation and the Academy of Finland project "Theory of Computational Logics" which is funded by grants 328987 and 324435.

References

Alechina, N.; Logan, B.; Nguyen, H.; and Rakib, A. 2009. A logic for coalitions with bounded resources. In *IJCAI'09*, 659–664.

Alechina, N.; Bulling, N.; Logan, B.; and Nguyen, H. 2015. On the boundary of (un)decidability: Decidable model-checking for a fragment of resource agent logic. In *IJCAI'15*, 1494–1501. AAAI Press.

Alechina, N.; Logan, B.; Nguyen, H.; and Raimondi, F. 2017. Model-checking for resource-bounded ATL with production and consumption of resources. *JCSS* 88:126–144.

Alechina, N.; Bulling, N.; Demri, S.; and Logan, B. 2018. On the complexity of resource-bounded logics. *TCS* 750:69–100.

Alur, R., and La Torre, S. 2001. Deterministic generators and games for LTL fragments. In *LiCS'01*, 291–300.

Alur, R.; Henzinger, T.; and Kupferman, O. 2002. Alternating-time temporal logic. *JACM* 49(5):672–713.

Belardinelli, F., and Demri, S. 2021. Strategic reasoning with a bounded number of resources: the quest for tractability. *AI* 300:103557.

Bouyer, P.; Fahrenberg, U.; Larsen, K.; Markey, N.; and Srba, J. 2008. Infinite runs in weighted timed automata with energy constraints. In *FORMATS'08*, volume 5215 of *LNCS*, 33–47. Springer.

Brázdil, T.; Jancar, P.; and Kucera, A. 2010. Reachability games on extended vector addition systems with states. In *ICALP'10*, volume 6199 of *LNCS*, 478–489. Springer.

Bruyère, V., and Raskin, J. 2007. Real-time model-checking: Parameters everywhere. *LMCS* 3(1).

Bulling, N., and Farwer, B. 2009. Expressing properties of resource-bounded systems: The logics RTL* and RTL. In *CLIMA X*, volume 6214 of *LNCS*, 22–45. Springer.

Bulling, N., and Farwer, B. 2010. On the (un-)decidability of model checking resource-bounded agents. In *ECAI'10*, 567–5–72. IOS Press.

Bulling, N., and Goranko, V. 2022. Combining quantitative and qualitative reasoning in concurrent multi-player games. *Autonomous Agents and Multi-Agent Systems* 36(2):1–33.

Bulling, N., and Jamroga, W. 2010. Verifying agents with memory is harder than it seemed. *AI Communications* 23(4):389–403.

Cerrito, S.; David, A.; and Goranko, V. 2015. Optimal tableau method for constructive satisfiability testing and model synthesis in the alternating-time temporal logic ATL^{+}. *ACM ToCL* 17(1):4:1–4:34.

Cerrito, S. 2021. Tableaux like model-checking on-the-fly for ATL+. arXiv:2107.05378.

Chatterjee, K.; Doyen, L.; and Henzinger, T. 2017. The cost of exactness in quantitative reachability. In *Models, Algorithms, Logics and Tools - Essays Dedicated to Kim Guldstrand Larsen on the Occasion of His 60th Birthday*, volume 10460 of *LNCS*, 367–381. Springer.

Chatterjee, K.; Henzinger, T.; and Piterman, N. 2008. Algorithms for Büchi Games. *CoRR* abs/0805.2620.

Chatterjee, K.; Henzinger, T.; and Piterman, N. 2010. Strategy logic. *I&C* 208(6):677–693.

Colcombet, T.; Jurdziński, M.; Lazić, R.; and Schmitz, S. 2017. Perfect half space games. In *LiCS'17*, 1–11. IEEE.

Czerwinski, W.; Lasota, S.; Lazić, R.; Leroux, J.; and Mazowiecki, F. 2020. Reachability in Fixed Dimension Vector Addition Systems with States. In *CONCUR'20*, volume 171 of *LIPIcs*, 48:1–48:21.

Demri, S.; Jurdziński, M.; Lachish, O.; and Lazić, R. 2013. The covering and boundedness problems for branching vector addition systems. *JCSS* 79(1):23–38.

Goranko, V., and Vester, S. 2014. Optimal decision procedures for satisfiability in fragments of alternating-time temporal logics. In *AiML'14*, 234–253. College Publications.

Goranko, V.; Kuusisto, A.; and Rönnholm, R. 2017. Game-theoretic semantics for ATL+ with applications to model checking. In *AAMAS'17*, 1277–1285. ACM.

Goranko, V.; Kuusisto, A.; and Rönnholm, R. 2018. Game-theoretic semantics for alternating-time temporal logic. *ACM ToCL* 19(3):17:1–17:38.

Goranko, V.; Kuusisto, A.; and Rönnholm, R. 2021. Game-theoretic semantics for ATL+ with applications to model checking. *I&C* 276.

Hélouët, L.; Markey, N.; and Raha, R. 2019. Reachability Games with Relaxed Energy Constraints. In *GandALF'19*, volume 305 of *EPTCS*, 17–33.

Jurdziński, M.; Lazić, R.; and Schmitz, S. 2015. Fixed-dimensional energy games are in pseudo-polynomial time. In *ICALP'15*, volume 9135 of *LNCS*, 260–272. Springer.

Karp, R., and Miller, R. E. 1969. Parallel program schemata. *JCSS* 3(2):147–195.

Laroussinie, F.; Markey, N.; and Oreiby, G. 2006. Model-checking timed ATL for durational concurrent game structures. In *FORMATS'06*, volume 4202 of *LNCS*, 245–259. Springer.

Laroussinie, F.; Markey, N.; and Oreiby, G. 2008. On the expressiveness and complexity of ATL. *LMCS* 4(2).

Leroux, J., and Schmitz, S. 2019. Reachability in Vector Addition Systems is Primitive-Recursive in Fixed Dimension. In *LiCS'19*, 1–13. IEEE.

Rackoff, C. 1978. The covering and boundedness problems for vector addition systems. *TCS* 6(2):223–231.

Vester, S. 2014. Model-checking quantitative alternating-time temporal logic on one-counter game models. *CoRR* abs/1409.5681.

Wang, F.; Schewe, S.; and Huang, C. 2015. An extension of ATL with strategy interaction. *ACM Trans. Program. Lang. Syst.* 37(3):9:1–9:41.