# Weighted Strategy Logic with Boolean Goals Over One-Counter Games*

**Patricia Bouyer, Patrick Gardy, and Nicolas Markey**

**LSV – CNRS, ENS Cachan, Université Paris-Saclay, France**
firstname.lastname@lsv.fr

## Abstract

Strategy Logic is a powerful specification language for expressing non-zero-sum properties of multi-player games. SL conveniently extends the logic ATL with explicit quantification and assignment of strategies. In this paper, we consider games over one-counter automata, and a quantitative extension 1cSL of SL with assertions over the value of the counter. We prove two results: we first show that, if decidable, model checking the so-called *Boolean-goal* fragment of 1cSL has non-elementary complexity; we actually prove the result for the Boolean-goal fragment of SL over finite-state games, which was an open question in [32]. As a first step towards proving decidability, we then show that the Boolean-goal fragment of 1cSL over one-counter games enjoys a nice periodicity property.

**1998 ACM Subject Classification** F.4.1. Mathematical Logic

**Keywords and phrases** Temporal logics, multi-player games, strategy logic, quantitative games

**Digital Object Identifier** 10.4230/LIPIcs.FSTTCS.2015.69

## 1 Introduction

**Model checking.** Model checking [19] has been developed for almost 40 years as a formal method for verifying correctness of computerized systems: this technique first consists in representing the system under study as a mathematical model (a finite-state transition system (a.k.a. Kripke structure), in the most basic setting), expressing the correctness property in some logical formalism (usually, using various *temporal logics* such as LTL [35] or CTL [18, 36]), and running an algorithm that exhaustively explores the set of behaviours of the model for proving or disproving the property.

Over the years, model checking has been extended in various directions, in order to take into account richer models and more precise properties. Several families of quantitative models (e.g. weighted Kripke structures [12], counter automata [25], timed automata [1]) and temporal logics [29, 24, 2, 7, 9, among others] have been defined and studied. Those formalisms conveniently extend the qualitative setting; they provide powerful ways of representing quantities, while in several cases keeping reasonably efficient model-checking algorithms.

Multi-agent systems (a.k.a. graph games [42, 4]) form another direction where model checking has been extended for reasoning about the interactions between components of a computerized system. Temporal logics have been extended accordingly [3, 16, 34, 20], in order to express the existence of *winning strategies* in multi-player games. Among the most popular approaches, the logic ATL [3] has limited expressive power but enjoys reasonably efficient model-checking algorithms, while the more expressive Strategy Logic (SL) [16, 34] extends LTL with explicit manipulation of strategies, and can express very rich non-zero-sum

* This work was partially supported by FP7 projects Cassting (601148) and ERC EQualIS (308087).


35th IARCS Annual Conf. Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2015).
Editors: Prahladh Harsha and G. Ramalingam; pp. 69–83
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

properties of games, including equilibria; however, model checking SL is non-elementary. Several fragments of SL have recently been introduced in order to mitigate the complexity of the model-checking problem while retaining the interesting aspects of SL [33, 13].

Quantitative games, combining both extensions, have also been widely considered. This includes games on weighted graphs [23, 14, 31, 8], games on counter systems or VASS [39, 11], or timed games [5, 21]. A large part of these works have focused on "simple" objectives, such as mean-payoff objectives [23], energy constraints [14, 8], or combinations thereof [15, 26].

**Our contribution.** In this paper, we consider a quantitative extension of SL over quantitative games. While such extensions have already been proven decidable for ATL [31, 43], we focus here on a quantitative extension of the richer logic SL, more specifically, its so-called *Boolean-goal* fragment SL[BG] [32]. SL with Boolean goals restricts SL by preventing arbitrary nesting of strategy quantifiers within temporal modalities. This and several other fragments of SL have been introduced in [32] with the aim of getting more efficient model-checking algorithms. However, while several fragments have been shown to have *2*-EXPTIME model-checking algorithms, the exact complexity of SL[BG] remained open.

We prove that model checking (the flat fragment of) SL[BG] is Tower-complete, thus negatively answering the open question whether SL[BG] would enjoy more efficient model-checking algorithm than SL. This hardness result obviously extends to the quantitative version 1cSL[BG] of SL[BG] over one-counter games. On the way to proving decidability of the model-checking problem for this logic, we then show that 1cSL[BG] over one-counter games enjoys a nice periodicity property: for any given formula, there is a threshold above which truth value of the formula is periodic (w.r.t. the value of the counter).

**Related works.** Several works have focused on one-counter models: two-player games with parity objectives have been proven PSPACE-complete [39]; this was recently extended to a quantitative extension of ATL [43], which is thus closely related to our paper. Model checking LTL and CTL over one-counter automata is also PSPACE-complete [28, 27]. Quantitative extensions of those logics have been studied in [22, 7, 9]. In many cases, they lead to undecidability of the model-checking problem. Games on VASS have also been considered, but reachability is only decidable in restricted cases [11, 37].

Games over integer-weighted graphs have a different flavour, as the behaviours do not depend on the value of the accumulated weight. Those games have been extensively considered with various quantitative objectives (e.g. mean-payoff [23, 44], energy [14, 8], and combinations thereof [15, 17]), and with objectives expressed in temporal logics [31, 6].

## 2 Definitions

▶ **Definition 1.** Let AP be a set of atomic propositions, and Agt be a set of agents. A *1-counter concurrent game structure* (1cCGS for short) is a tuple $\mathcal{G} = \langle \mathsf{Loc}, \mathsf{label}, \mathsf{Act}, \mathsf{Tab}_{\{0,1\}}, \mathsf{Wgt}_{\{0,1\}} \rangle$ where

- $\mathsf{Loc}$ is a finite set of locations;
- $\mathsf{label}\colon \mathsf{Loc} \to 2^{\mathsf{AP}}$ labels locations with atomic propositions;
- $\mathsf{Act}$ is a finite set of actions;
- $\mathsf{Tab}_0\colon \mathsf{Loc} \times \mathsf{Act}^{\mathsf{Agt}} \to \mathsf{Loc}$ and $\mathsf{Tab}_1\colon \mathsf{Loc} \times \mathsf{Act}^{\mathsf{Agt}} \to \mathsf{Loc}$ are two transition tables;
- $\mathsf{Wgt}_0\colon \mathsf{Loc} \times \mathsf{Act}^{\mathsf{Agt}} \to \{0,1\}$ and $\mathsf{Wgt}_1\colon \mathsf{Loc} \times \mathsf{Act}^{\mathsf{Agt}} \to \{-1,0,1\}$ assign a weight to each transition of the transition tables.

A *finite path* in a 1cCGS $\mathcal{G}$ is a finite non-empty sequence of configurations $\rho = \gamma_0\gamma_1\gamma_2\ldots\gamma_k$, where for all $0 \le i \le k$, the configuration $\gamma_i$ is a pair $(\ell_i, c_i)$ with $\ell_i \in \mathsf{Loc}$ and $c_i \in \mathbb{N}$. For such a path, we denote by $\mathsf{last}(\rho)$ its last element $\gamma_k$, and we let $|\rho| = k$. number of transitions An *infinite path* is an infinite sequence of configurations with the same property. We denote by Path (resp. InfPath) the set of finite (resp. infinite) paths. The length of an infinite path is $+\infty$. For $0 \le i < |\rho|$, $\rho(i)$ represents the $i+1$-th element $\gamma_i$ of $\rho$. For a path $\rho$ and $0 \le i < |\rho|$, we denote by $\rho_{\le i}$ the prefix of $\rho$ until position $i$, i.e. the finite path $\rho(0)\rho(1)\ldots\rho(i)$.

A *strategy* for some agent $a \in \mathsf{Agt}$ is a function $\sigma_a\colon \mathsf{Path} \to \mathsf{Act}$. We write Strat for the set of strategies. Given a finite path (or *history*) in the game, a strategy $\sigma_a$ returns the action that agent $a$ will play next. A strategy $\sigma_A$ for a coalition of agents $A \subseteq \mathsf{Agt}$ is a function assigning a strategy $\sigma_A(a)$ to each agent $a \in A$. Given a strategy $\sigma_A$ for coalition $A$, we say that a path $\rho$ respects $\sigma_A$ after a finite prefix $\pi$ if, writing $\rho(i) = (\ell_i, c_i)$ for all $0 \le i \le |\rho|$, the following two conditions hold:

- for all $0 \le i < |\pi|$, we have $\rho(i) = \pi(i)$
- for all $|\pi| \le i < |\rho| - 1$, we have that $\ell_{i+1} = \mathsf{Tab}_s(\ell_i, \mathbf{m})$ and $c_{i+1} = c_i + \mathsf{Wgt}_s(\rho_{\le i}, \mathbf{m})$, where $s = 0$ if $c_i = 0$ and $s = 1$ otherwise, and $\mathbf{m}$ is an action vector satisfying $\mathbf{m}(a) = \sigma_A(a)(\rho_{\le_i})$ for all $a \in A$.

Notice that the value of the counter always remains nonnegative as $\mathsf{Wgt}_0$ only returns nonnegative values. Given a finite path $\pi$, we denote by $\mathsf{Out}(\pi, \sigma_A)$ the set of paths that respect the strategy $\sigma_A$ after prefix $\pi$. Notice that if $\sigma_A$ assigns a strategy to all the agents, then $\mathsf{Out}(\pi, \sigma_A)$ contains a single path, which we write $\mathsf{out}(\pi, \sigma_A)$.

▶ Remark. Several semantics have been given to quantitative games, see [37]. The semantics chosen here, with zero tests (using $\mathsf{Tab}_0, \mathsf{Tab}_1$), allows to easily express the three semantics studied in [37]. Hence our algorithms apply in all these settings. It is worth noticing that the hardness proof holds for the non-quantitative setting, hence also for all three semantics mentioned above.

We now define our weighted extension of Strategy Logic [16, 34]:

▶ **Definition 2.** Let AP be a set of atomic propositions, Agt be a set of agents, and Var be a finite set of strategy variables. Formulas in 1cSL are built from the following grammar:

$$\mathsf{1cSL} \ni \phi ::= p \mid \mathsf{cnt} \in S \mid \neg\,\phi \mid \phi \vee \phi \mid \mathbf{X}\,\phi \mid \phi\,\mathbf{U}\,\phi \mid \exists x.\ \phi \mid \mathsf{bind}(a \mapsto x).\ \phi$$

where $p$ ranges over AP, $S$ is a subset of $\mathbb{N}$ that can be described as $S^1_{\mathsf{fin}} \cup \big(S^2_{\mathsf{fin}} + k \cdot \mathbb{N}\big)$, where $S^i_{\mathsf{fin}}$ are finite subsets of $\mathbb{N}$ and $k \in \mathbb{N}$ is a period[1], $x$ ranges over Var, and $a$ ranges over Agt. The logic SL is the fragment of 1cSL where no counter constraint $\mathsf{cnt} \in S$ or $\mathsf{cnt} \in S_{[k]}$ is used. The logic 1cLTL is the fragment of 1cSL where no strategy quantifiers $\exists x.\ \phi$ and no strategy bindings $\mathsf{bind}(a \mapsto x).\ \phi$ are used. Finally, LTL is the intersection of SL and 1cLTL.

The set of *free agents and variables* of a formula $\phi$ of 1cSL, which we write $\mathsf{free}(\phi)$, contains the agents and variables that have to be associated with a strategy before $\phi$ can be

[1] This allows to express standard counter constraints like $\mathsf{cnt} \ge 5$ (using negation) or periodic constraint like $\mathsf{cnt} = 4 \bmod 7$. Notice that our periodicity result is not a consequence of the periodicity of the quantitative assertions, and would also hold with assertions of the form $\mathsf{cnt} \sim n$.

evaluated. It is defined inductively as follows:

$$
\begin{array}{rlrl}
\mathsf{free}(p) = \varnothing & \text{for } p \in \mathsf{AP} & \mathsf{free}(\mathbf{X}\,\phi) = & \mathsf{Agt} \cup \mathsf{free}(\phi) \\
\mathsf{free}(\mathsf{cnt} \in S) = \varnothing & \text{for } n \in \mathbb{N} & \mathsf{free}(\phi\,\mathbf{U}\,\psi) = & \mathsf{Agt} \cup \mathsf{free}(\phi) \cup \mathsf{free}(\psi) \\
\mathsf{free}(\neg\,\phi) = \mathsf{free}(\phi) & & \mathsf{free}(\phi \vee \psi) = & \mathsf{free}(\phi) \cup \mathsf{free}(\psi) \\
\mathsf{free}(\exists x.\ \phi) = \mathsf{free}(\phi) \setminus \{x\} & & \mathsf{free}(\mathsf{bind}(a \mapsto x).\ \phi) = & \begin{cases} \mathsf{free}(\phi) & \text{if } a \notin \mathsf{free}(\phi) \\ (\mathsf{free}(\phi) \cup \{x\}) \setminus \{a\} & \text{otherwise} \end{cases}
\end{array}
$$

A formula $\phi$ is *closed* if $\mathsf{free}(\phi) = \emptyset$.

We can now define the semantics of 1cSL. Let $\mathcal{G}$ be a 1cCGS, $\pi$ be a path, $i$ be a position along $\pi$, and $\chi\colon \mathsf{Var} \cup \mathsf{Agt} \dashrightarrow \mathsf{Strat}$ be a partial valuation (or context) with domain $\mathsf{dom}(\chi)$. Let $\phi \in \mathsf{SL}$ such that $\mathsf{free}(\phi) \subseteq \mathsf{dom}(\chi)$. Whether $\phi$ holds true at position $i$ along $\pi$ within context $\chi$ is defined inductively as follows:

$$
\begin{array}{lll}
\mathcal{G}, \pi, i \models_\chi p & \text{iff} & p \in \mathsf{label}(\ell_i) \qquad (\text{writing } \pi(i) = (\ell_i, c_i)) \\
\mathcal{G}, \pi, i \models_\chi \mathsf{cnt} \in S & \text{iff} & c_i \in S \qquad (\text{writing } \pi(i) = (\ell_i, c_i)) \\
\mathcal{G}, \pi, i \models_\chi \neg\,\phi_1 & \text{iff} & \mathcal{G}, \pi, i \not\models_\chi \phi_1 \\
\mathcal{G}, \pi, i \models_\chi \phi_1 \vee \phi_2 & \text{iff} & \mathcal{G}, \pi, i \models_\chi \phi_1 \ \text{ or } \ \mathcal{G}, \pi, i \models_\chi \phi_2 \\
\mathcal{G}, \pi, i \models_\chi \mathbf{X}\,\phi_1 & \text{iff} & \mathcal{G}, \rho, i+1 \models_\chi \phi_1 \qquad (\text{writing } \rho = \mathsf{out}(\pi_{\le i}, \chi_{|\mathsf{Agt}})) \\
\mathcal{G}, \pi, i \models_\chi \phi_1\,\mathbf{U}\,\phi_2 & \text{iff} & \exists k \ge i.\ \mathcal{G}, \rho, k \models_\chi \phi_2 \text{ and} \\
 & & \forall i \le j < k.\ \mathcal{G}, \rho, j \models_\chi \phi_1 \qquad (\text{writing } \rho = \mathsf{out}(\pi_{\le i}, \chi_{|\mathsf{Agt}})) \\
\mathcal{G}, \pi, i \models_\chi \exists x.\ \phi_1 & \text{iff} & \exists \sigma \in \mathsf{Strat}.\ \mathcal{G}, \pi, i \models_{\chi[x \mapsto \sigma]} \phi_1 \\
\mathcal{G}, \pi, i \models_\chi \mathsf{bind}(a \mapsto x).\ \phi_1 & \text{iff} & \mathcal{G}, \pi, i \models_{\chi[a \mapsto \chi(x)]} \phi_1
\end{array}
$$

Notice that the constraint that $\mathsf{free}(\phi) \subseteq \mathsf{dom}(\chi)$ is preserved at each step.

▶ Remark. One may notice that the relation $\mathcal{G}, \pi, i \models_\chi \phi$ does not depend on the suffix of $\pi$ after position $i$. Moreover, writing $\sigma_{\overrightarrow{\pi_{\le i}}}$ for the strategy $\sigma'$ such that $\sigma'(\rho) = \sigma(\pi_{\le i} \cdot \rho)$, it is easily proved that $\mathcal{G}, \pi, i \models_\chi \phi$ if, and only if, $\mathcal{G}, \pi', 0 \models_{\chi'} \phi$, where $\chi'(x) = \chi(x)_{\overrightarrow{\pi_{\le i}}}$ for all $x \in \mathsf{Var} \cup \mathsf{Agt}$ (we will later write $\chi_{\overrightarrow{\pi_{\le i}}}$ for $\chi'$). As the satisfaction relation does not depend on the suffix of $\pi$ after position $i$, we may also write $\mathcal{G}, \gamma \models_{\chi'} \phi$, where $\gamma = \pi(i)$. In the sequel, we may even omit to mention $\mathcal{G}$ when it is clear from the context, and simply write $\gamma \models_\chi \phi$.

▶ Remark. We write $\langle\!\cdot a \cdot\!\rangle\, \phi$ as a shorthand for $\exists \sigma_a.\ \mathsf{bind}(a \mapsto \sigma_a).\ \phi$, when we do not need to have hands on $\sigma_a$ in the rest of the formula. Similarly, $[\cdot a \cdot]\, \phi$ stands for $\neg \langle\!\cdot a \cdot\!\rangle\, \neg\,\phi$. This construct $\langle\!\cdot a \cdot\!\rangle\, \phi$ precisely corresponds to the strategy quantification used in the logic $\mathsf{ATL}_{sc}$ [30], but it should be noticed that it does *not* correspond to the strategy quantifier of ATL [3].

In the sequel, we also use other classical shorthands such as $\top$, defined as $p \vee \neg\, p$ for some $p$ (hence it is always true); $\mathbf{F}\,\phi$ as a shorthand for $\top\,\mathbf{U}\,\phi$, meaning that $\phi$ holds at a later position; and $\mathbf{G}\,\phi$, defined as $\neg\,\mathbf{F}\,\neg\,\phi$, meaning that $\phi$ holds true at every future position.

Several fragments of SL have recently been defined and studied [32]. Those fragments restrict the use of strategy bindings and quantifications. In the present paper, we are mainly interested in the quantitative extension of the fragment SL[BG]. Before defining 1cSL[BG], we first introduce its *flat* fragment $\mathsf{1cSL}^0$[BG]:

$$
\begin{array}{rl}
\mathsf{1cSL}^0[\mathsf{BG}] \ni \phi ::= & \neg\,\phi \mid \phi \vee \phi \mid \exists x.\ \phi \mid \mathsf{bind}(a \mapsto x).\ \phi \mid \psi \\
\psi ::= & p \mid \mathsf{cnt} \in S \mid \neg\,\psi \mid \psi \vee \psi \mid \mathbf{X}\,\psi \mid \psi\,\mathbf{U}\,\psi
\end{array}
$$

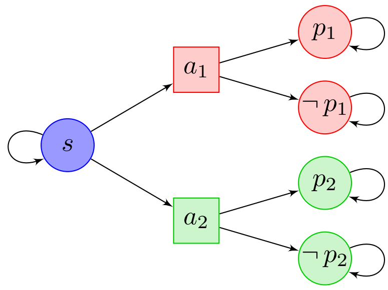



**Figure 1** The 3-player turn-based game for the reduction to SL model checking.

▶ Remark. Any closed formula $\varphi$ in $\mathsf{1cSL}^0[\mathsf{BG}]$ can be written in *prenex form* as

$$\wp(\mathsf{Var}).\ f\Big((\beta_i(\mathsf{Agt},\mathsf{Var}).\ \psi_i)_{1\leq i\leq n}\Big)$$

where $\wp(\mathsf{Var})$ is a series of strategy quantifiers involving all variables in $\mathsf{Var}$, $f$ is a Boolean combination over $n$ atoms, and for every $1 \leq i \leq n$, $\beta_i$ assigns a strategy from $\mathsf{Var}$ to each agent of $\mathsf{Agt}$, and $\psi_i$ is a $\mathsf{1cLTL}$ formula.

$\mathsf{1cSL}[\mathsf{BG}]$ then extends $\mathsf{1cSL}^0[\mathsf{BG}]$ by allowing nesting *closed* formulas at the level of atomic propositions. Formally, we defined the depth-$i$ fragment as

$$\begin{aligned}\mathsf{1cSL}^i[\mathsf{BG}] \ni \phi &::= \neg\,\phi \mid \phi \vee \phi \mid \exists x.\ \phi \mid \mathsf{bind}(a \mapsto x).\ \phi \mid \psi \\ \psi &::= p \mid \phi_{i-1} \mid \mathsf{cnt} \in S \mid \neg\,\psi \mid \psi \vee \psi \mid \mathbf{X}\,\psi \mid \psi\,\mathbf{U}\,\psi\end{aligned}$$

where $\phi_{i-1}$ ranges over closed formulas of $\mathsf{1cSL}^{i-1}[\mathsf{BG}]$. We let $\mathsf{1cSL}[\mathsf{BG}]$ be the union of the fragments $\mathsf{1cSL}^i[\mathsf{BG}]$ for all $i \in \mathbb{N}$. It can be checked that if we drop the quantitative constraints from $\mathsf{1cSL}[\mathsf{BG}]$, we precisely get the logic $\mathsf{SL}[\mathsf{BG}]$ of [32].

# 3 Hardness of SL[BG] model checking

In this section, we prove that the model-checking problem for SL[BG] is Tower-hard (the complexity class Tower is the union of all classes $k$-EXPTIME when $k$ ranges over $\mathbb{N}$ [38]). We actually prove the result for (the flat fragment of) SL[BG], closing a question left open in [32].

▶ **Theorem 3.** *Model checking SL[BG], and hence 1cSL[BG], is* Tower*-hard.*

We give a sketch of the proof here, and develop the full proof in [10].

**Sketch of proof.** We prove this result by encoding the satisfiability problem for QLTL into the model-checking problem for SL[BG]. QLTL is the extension of LTL with quantification over atomic propositions [40]: formulas in QLTL are of the form $\Phi = \forall p_1 \exists p_2 \ldots \forall p_{n-1} \exists p_n.\ \varphi$ where $\phi$ is in LTL. Notice that we only consider strictly alternating formulas for the sake of readability. The general case can be handled similarly. Formula $\exists p.\ \varphi$ holds true over a word $w \colon \mathbb{N} \to 2^{\mathsf{AP}}$ if there exists a word $w' \colon \mathbb{N} \to 2^{\mathsf{AP}}$ with $w'(i) \cap (\mathsf{AP} \setminus \{p\}) = w(i) \cap (\mathsf{AP} \setminus \{p\})$ and $w' \models \varphi$ for all $i$. Universal quantification is defined similarly. It is well-known that model checking (and satisfiability) of QLTL is Tower-complete [41]. We reduce the satisfiability of QLTL into a model-checking problem for a SL[BG] formula involving $n+4$ players (where $n$ is the number of quantifiers in the QLTL formula), and three additional quantifier alternations.

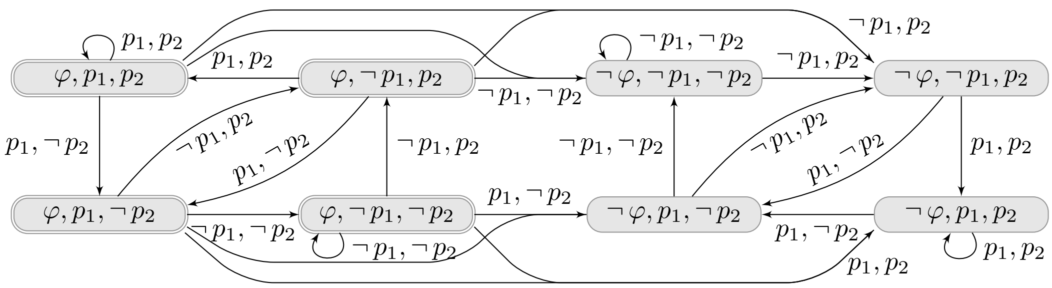


**Figure 2** Büchi automaton for $\mathbf{G}\,(p_2 \Leftrightarrow \mathbf{X}\,p_1)$.

Before developing this technical encoding, we first present an example of a reduction to plain SL, which already contains most of the intuitions of our reduction to SL[BG]. Consider the QLTL formula

$$\Phi = \forall p_1.\ \exists p_2.\ \mathbf{G}\,(p_2 \Leftrightarrow \mathbf{X}\,p_1).$$

To solve the satisfiability problem of this formula via SL, we use the three-player turn-based game depicted on Fig. 1. In that game, Player Blue controls the blue state $s$, while Players Red and Green control the square states $a_1$ and $a_2$, respectively. Fix a strategy of Player Red: this strategy will be evaluated only in red state $a_1$, hence after histories of the form $s^n \cdot a_1$. Hence a strategy of Player Red can be seen as associating with each integer $n$ a value for $p_1$. In other words, a strategy for Player Red defines a labeling of the time line with atomic proposition $p_1$. Similarly for Player Green and proposition $p_2$.

It remains to use this correspondence for encoding our QLTL formula. We have to express that for any strategy $\sigma_{\mathsf{Red}}$ of Player Red, there is a strategy $\sigma_{\mathsf{Green}}$ of Player Green under which, at each step along the path that stays in $s$ forever, Player Blue can enforce $\mathbf{X}\,\mathbf{X}\,p_2$ if, and only if, he can enforce $\mathbf{X}\,\mathbf{X}\,p_1$ one step later. In the end, the formula reads as follows:

$$[\cdot\mathsf{Red}\cdot]\ \langle\cdot\mathsf{Green}\cdot\rangle\ \langle\cdot\mathsf{Blue}\cdot\rangle\ \mathbf{G}\,\Big(s \wedge (\ \langle\cdot\mathsf{Blue}\cdot\rangle\ \mathbf{X}\,\mathbf{X}\ p_2\ ) \Leftrightarrow (\mathbf{X}\ \langle\cdot\mathsf{Blue}\cdot\rangle\ \mathbf{X}\,\mathbf{X}\ p_1\ )\Big) \tag{1}$$

One may notice that the above property is not in SL[BG]: for instance, the subformula $\langle\cdot\mathsf{Blue}\cdot\rangle\,\mathbf{X}\,\mathbf{X}\ p_2$ is not closed. We provide a different construction, refining the ideas above, in order to reduce QLTL satisfiability to SL[BG] model checking.

In order to do so, we take another approach for encoding the LTL formula, since our technique of encoding $p_i$ with $\langle\cdot\mathsf{Blue}\cdot\rangle\,\mathbf{X}\,\mathbf{X}\ p_i$ is not compatible with getting a formula in SL[BG]. Instead, we will use a Büchi automaton encoding the formula; another player, say Player Black, will be in charge of selecting states of the Büchi automaton at each step. Using the same trick as above in the game structure on the left of Fig. 3, a strategy for Player Black can be seen as a mapping from $\mathbb{N}$ to states of the Büchi automaton. Our formula will ensure that this sequence of states is in accordance with the atomic propositions selected by the square players in states $a_i$, and that it forms an accepting run of the Büchi automaton.

For our example, an eight-state Büchi automaton associated with the (LTL part of the) QLTL formula is depicted on Fig. 2. Notice that smaller automata exist for this property (for instance, the four states on the right could be merged into a single one), but for technical reasons in our construction, we require that each state of the Büchi automaton corresponds to a single valuation of the atomic propositions, hence the number of states must be a multiple of $2^{|\mathsf{AP}|}$. Accordingly, we augment our game structure of Fig. 1 with eight extra states, as depicted on the left of Fig. 3. Again, a strategy of Player Black (controlling state $b$) defines a sequence of states of the Büchi automaton.

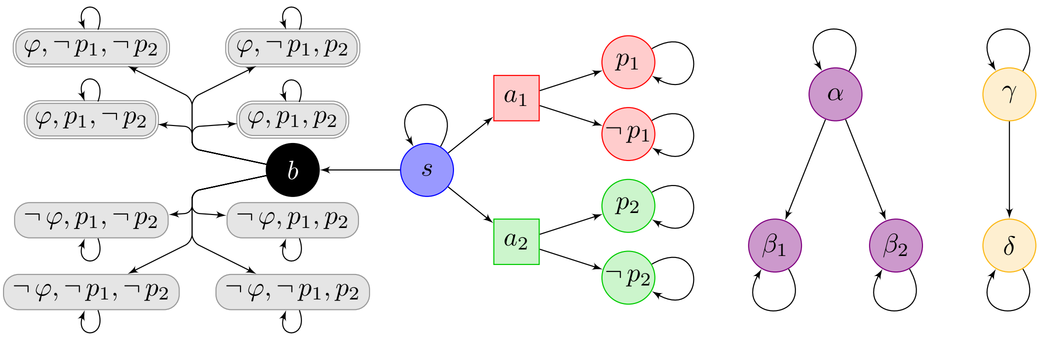



**Figure 3** The concurrent game for the reduction to SL[BG] model checking.

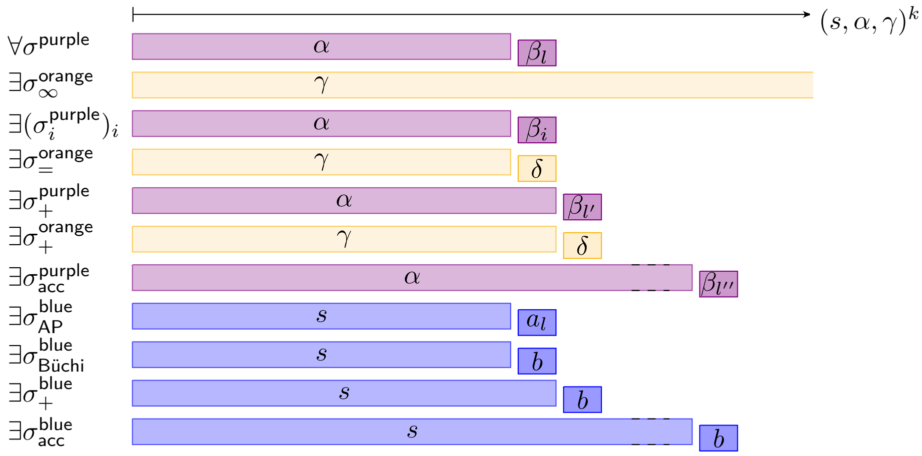



**Figure 4** Visualization of the strategies selected by $\Psi_{\mathsf{aux}}$ on history $(s, \alpha, \gamma)^k$.

It then remains to "synchronize" the run of the Büchi automaton with the valuations of the atomic propositions, selected by the players controlling the square states. This is achieved by taking the product of the game we just built with two extra one-player structures, as depicted on the right of Fig. 3. The product gives rise to a concurrent game, where one transition is taken simultaneously in the main structure and in the Purple and Orange structures. In this product, as long as Player Blue remains in $s$ and Player Purple remains in $\alpha$, a strategy of Player Orange (controlling state $\gamma$) either remains in $\gamma$ forever, or it can be characterized by a value $n \in \mathbb{N}$. Similarly, as long as Player Blue remains in $s$ and Player Orange remains in $\gamma$, a strategy of Player Purple (controlling state $\alpha$) either loops forever in $\alpha$, or can be uniquely characterized by a pair $(k, p_l)$, where $k$ is the number of times the loop over $\alpha$ is taken before entering state $\beta_l$ corresponding to $p_l \in \mathsf{AP}$.

Our construction can then be divided in two steps:

- First, with any strategy of Player Purple (characterized by $(k, p_l)$ for the interesting cases), we associate auxiliary strategies of Players Blue, Purple and Orange satisfying certain properties, that can be enforced by an SL[BG] formula $\Psi_{\mathsf{aux}}$; Fig. 4 should help visualizing the associated strategies; in particular, strategies $\sigma_+^{\mathsf{orange}}$, $\sigma_+^{\mathsf{blue}}$ and $\sigma_+^{\mathsf{purple}}$ characterize position $k+1$ (which will be useful for checking transitions of the Büchi automaton), while $\sigma_{\mathsf{Büchi}}^{\mathsf{blue}}$ and $\sigma_{\mathsf{AP}}^{\mathsf{blue}}$ are Player-Blue strategies that either go to the Büchi part or to the proposition part of the main part of the game.

- Then, using those strategies, we write another $\mathsf{SL[BG]}$ formula to enforce that the transitions of the Büchi automaton are correctly applied, following the valuations of the atomic propositions selected in the square states, and that an accepting state is visited infinitely many times.

The construction of the game structure $\mathcal{G}_\Phi$ depicted on Fig. 3 is readily extended to any number of atomic propositions, and to any Büchi automaton. We now explain how we build our $\mathsf{SL[BG]}$ formula replacing Formula (1), and ensuring correctness of our reduction.

We do not detail the first step mentioned above and assume that a formula $\Psi_{\mathsf{aux}}$ has been written, which properly generates auxiliary strategies, as depicted on Fig. 4 (see [10]). Instead we focus on the Büchi automaton simulation. We look for a strategy of Player Black that will mimic the run of the Büchi automaton, following the valuation of the atomic propositions selected by the square players $A_1$ to $A_n$. We also require that the run of the Büchi automaton be accepting.

The formula $\Psi$ enforcing these constraints is as follows[2]:

$$\forall\sigma^{A_1}.\ \exists\sigma^{A_2}.\ \ldots\forall\sigma^{A_{n-1}}.\ \exists\sigma^{A_n}.\ \exists\sigma^{\mathsf{black}}.\ \mathsf{bind}(\sigma^{A_1},\sigma^{A_2},\ldots,\sigma^{A_{n-1}},\sigma^{A_n},\sigma^{\mathsf{black}},\sigma^{\mathsf{orange}}_{\infty}).\ \Psi_{\mathsf{aux}}$$
$$\wedge \bigwedge_{p_i,p_j\in\mathsf{AP}}\ \bigwedge_{q\in Q} (\mathsf{bind}(\sigma^{\mathsf{blue}}_{\mathsf{Büchi}},\sigma^{\mathsf{purple}}_{i})\mathbf{F}\,q) \Leftrightarrow (\mathsf{bind}(\sigma^{\mathsf{blue}}_{\mathsf{Büchi}},\sigma^{\mathsf{purple}}_{j})\mathbf{F}\,q) \qquad (\varphi_1)$$
$$\wedge \bigwedge_{p_i\in\mathsf{AP}} \Big(\big(\mathsf{bind}(\sigma^{\mathsf{blue}}_{\mathsf{AP}},\sigma^{\mathsf{purple}}).\ \mathbf{F}\,p_i\big) \Rightarrow \big(\mathsf{bind}(\sigma^{\mathsf{blue}}_{\mathsf{Büchi}},\sigma^{\mathsf{purple}}).\ \bigvee_{q\in Q\mid p_i\in\mathsf{label}(q)} \mathbf{F}\,q\big)\Big) \qquad (\varphi_2)$$
$$\wedge \bigwedge_{p_i\in\mathsf{AP}} \Big(\big(\mathsf{bind}(\sigma^{\mathsf{blue}}_{\mathsf{AP}},\sigma^{\mathsf{purple}}).\ \mathbf{F}\,\neg p_i\big) \Rightarrow \big(\mathsf{bind}(\sigma^{\mathsf{blue}}_{\mathsf{Büchi}},\sigma^{\mathsf{purple}}).\ \bigvee_{q\in Q\mid p_i\notin\mathsf{label}(q)} \mathbf{F}\,q\big)\Big) \qquad (\varphi_3)$$
$$\wedge \bigwedge_{q\in Q} \mathsf{bind}(\sigma^{\mathsf{blue}}_{\mathsf{Büchi}},\sigma^{\mathsf{purple}}).\ \mathbf{F}\,q \Rightarrow \bigvee_{q'\in\mathsf{succ}(q)} \mathsf{bind}(\sigma^{\mathsf{blue}}_{+},\sigma^{\mathsf{purple}}_{+}).\ \mathbf{F}\,q' \qquad (\varphi_4)$$
$$\wedge\ \mathsf{bind}(\sigma^{\mathsf{blue}}_{\mathsf{acc}},\sigma^{\mathsf{purple}}_{\mathsf{acc}}).\ \bigvee_{q\in\mathsf{accept}(Q)} \mathbf{F}\,q \qquad (\varphi_5)$$

We now analyze formula $\Psi$:

- Formula $(\varphi_1)$ requires that strategy $\sigma^{\mathsf{black}}$ returns the same move after any history of the form $(s,\alpha,\gamma)^k(b,\beta_i,\gamma)$, whichever $\beta_i$ has been selected by $\sigma^{\mathsf{purple}}$;
- Formulas $(\varphi_2)$ and $(\varphi_3)$ constrain the state of the Büchi automaton to correspond to the valuation of the atomic propositions selected. Because of the universal quantification over $\sigma^{\mathsf{purple}}$, this property will be enforced at all positions and for all atomic propositions;
- Formula $(\varphi_4)$ additionally requires that two consecutive states of the run of the Büchi automaton indeed correspond to a transition;
- finally, Formula $(\varphi_5)$ states that for any position (selected by $\sigma^{\mathsf{purple}}$), there exists a later position (given by $\sigma^{\mathsf{purple}}_{\mathsf{acc}}$) at which the run of the Büchi automaton visits an accepting state.

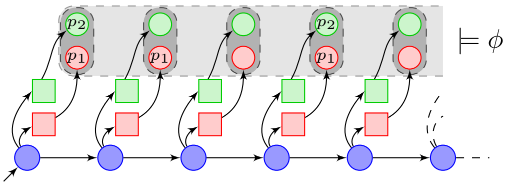
The correctness of the construction is then stated in the next lemma, whose proof can be found in [10].

[2] We notice that $\Psi$ is not syntactically in $\mathsf{SL[BG]}$, as some bindings appear before quantifications in $\Psi_{\mathsf{aux}}$. However, quantifiers in $\Psi_{\mathsf{aux}}$ could be moved before the bindings of $\Psi$.

▶ **Lemma 4.** *Formula $\Phi$ in* QLTL *is satisfiable if, and only if, Formula $\Psi$ in SL[BG] holds true in state $(s, \alpha, \gamma)$ of the game $\mathcal{G}_\Phi$.* ◀

▶ Remark. SL[BG] and several other fragments were defined in [32, 33] with the aim of getting more tractable fragments of SL. In particular, the authors advocate for the restriction to *behavioural strategies*: this forbids strategies that prescribe actions depending of what other strategies would prescribe later on, or after different histories. Non-behavioural strategies are thus claimed to have limited interest in practice; moreover, they are suspected of being responsible for the non-elementary complexity of SL model-checking. Our hardness result strengthens the latter claim, as SL[BG] is known for not having behavioral strategies.

▶ Remark. We had to rely on a Büchi automaton instead of directly using the original LTL formula directly in the SL[BG] formula. This is because we need to evaluate the formula not on a real path of our game structure, but on a sequence of "unions" of states. The figure on the right represents this situation for the game structure of Fig. 1: the path on which the LTL formula is given by the red and green circle states, which define the valuations for $p_1$ and $p_2$.




## 4 Periodicity of 1cSL[BG] model checking

In this section we prove our periodicity property for 1cSL[BG]. We inductively define the function $\mathsf{tower}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ as $\mathsf{tower}(a, 0) = a$ and $\mathsf{tower}(a, b+1) = 2^{\mathsf{tower}(a,b)}$. This encodes *towers of exponentials* of the form $2^{2^{\cdot^{\cdot^{\cdot^{a}}}}}$.

▶ **Theorem 5.** *Let $\mathcal{G}$ be a 1cCGS, and $\varphi$ be a 1cSL[BG] formula. Then there exist a threshold $h \geq 0$ and a period $\Lambda \geq 0$ for the truth value of $\varphi$ over $\mathcal{G}$. That is, for every configuration $(q, c)$ of $\mathcal{G}$ with $c \geq h$, for every $k \in \mathbb{N}$, $\mathcal{G}, (q, c) \models \varphi$ if, and only if, $\mathcal{G}, (q, c + k \cdot \Lambda) \models \varphi$.*

*Furthermore the order of magnitude for $h + \Lambda$ is bounded by*

$$\mathsf{tower}\left(\max_{\theta \in \mathsf{Subf}(\varphi)} n_\theta, \max_{\theta \in \mathsf{Subf}(\varphi)} k_\theta + 1\right)^{|Q| \cdot 2^{2^{|\varphi|}}}$$

*where $\mathsf{Subf}(\varphi)$ is the set of 1cSL[BG] formulas of $\varphi$, $k_\theta$ is the number of quantifier alternations in $\theta$, and $n_\theta$ is the number of different bindings used in $\theta$.*

The rest of this section is devoted to developing the proof of this result, though not with full details. Detailed proofs of intermediate results are given in [10].

We first prove this property for the flat fragment $\mathsf{1cSL}^0$[BG], and then extend it to the full 1cSL[BG].

### 4.1 The flat fragment $\mathsf{1cSL}^0$[BG]

We fix a 1cCGS $\mathcal{G}$ and a formula $\varphi = Q_1 x_1 \ldots Q_k x_k.\ f((\beta_i \phi_i)_{1 \leq i \leq n})$ in $\mathsf{1cSL}^0$[BG], where for every $1 \leq j \leq k$, we have $Q_j \in \{\exists, \forall\}$ (assuming quantifiers strictly alternate), $f$ is a Boolean formula over $n$ atoms, and for every $1 \leq i \leq n$, $\beta_i$ is a complete binding for the players' strategies, and $\phi_i$ is a 1cLTL formula. We write $M$ for the maximal constant appearing in one of the finite sets describing a counter constraint $S$ appearing in $\varphi$.

For every $1 \leq i \leq n$, we let $\mathcal{D}_i$ be a deterministic (counter) parity automaton that recognizes formula $\phi_i$ (this is the standard LTL-to-(deterministic parity) automata construction in which quantitative constraints are seen as atoms). A run of $\mathcal{G}$ is read in a standard way, with the additional condition that quantitative constraints labelling a state should be satisfied by the counter value when the state is traversed (a state can be labelled by a constraint $\mathsf{cnt} \in S$, with $S$ arbitrarily complex—it does not impact the description of the automaton).

The proof proceeds by showing that, above some threshold, the truth value of $\varphi$ is periodic w.r.t. counter values. To prove this, we define an equivalence relation over counter values that generates identical strategic possibilities (in a sense that will be made clear later on).

### 4.1.1 Definition of an equivalence relation

Fix a configuration $\gamma = (\ell, c)$ in $\mathcal{G}$, pick for every $1 \leq i \leq n$ a state $d_i$ in the automaton $\mathcal{D}_i$, and define the tuple $D = (d_1, \dots, d_n)$. For every context $\chi_k$ for variables $\{x_1, \dots, x_k\}$, we define the level-0 identifier $\mathsf{Id}_{\chi_k}(\gamma, D)$ as:

$$\mathsf{Id}_{\chi_k}(\gamma, D) = \left\{ i \;\middle|\; 1 \leq i \leq n \text{ and } \mathsf{out}(\gamma, \beta_i[\chi_k]) \text{ is accepted by } \mathcal{D}_i \text{ from } d_i \right\}$$

where $\beta_i[\chi_k]$ assigns a strategy from $\chi_k$ to each player in $\mathsf{Agt}$ following $\beta_i$.

Assuming we have defined level-$(k - j + 1)$ identifiers $\mathsf{Id}_{\chi_{j+1}}(\gamma, D)$ for every partial context $\chi_{j+1}$ for variables $\{x_1, \dots, x_{j+1}\}$, we define the level-$(k - j)$ identifier $\mathsf{Id}_{\chi_j}(\gamma, D)$ for every partial context $\chi_j$ for variables $\{x_1, \dots, x_j\}$ as follows:

$$\mathsf{Id}_{\chi_j}(\gamma, D) = \left\{ \mathsf{Id}_{\chi_{j+1}}(\gamma, D) \;\middle|\; \chi_{j+1} \text{ is a context for } \{x_1, \dots, x_{j+1}\} \text{ that extends } \chi_j \right\}.$$

There is a unique level-$k$ identifier for every configuration $\gamma = (\ell, c)$ and every $D$, which corresponds to the empty context. It somehow contains full information about what kinds of strategies can be used in the game (this is a hierarchical information set, which contains all level-$j$ identifiers for $j < k$).

Let $P$ be the least common multiple of all the periods appearing in periodic quantitative assertions used in formula $\varphi$. We define the following equivalence on counter values:

$$c \sim c' \quad \text{if, and only if,} \quad c = c' \bmod P \text{ and } \forall D.\ \forall \ell.\ \mathsf{Id}_\emptyset((\ell, c), D) = \mathsf{Id}_\emptyset((\ell, c'), D).$$

**Combinatorics.** Given a configuration $(\ell, c)$ and a tuple $D$, the number of possible values for the level-0 identifier is $\mathsf{tower}\,(n, 1)$, and for the level-$j$ identifier it is $\mathsf{tower}\,(n, j + 1)$. Hence, the number $\mathsf{ind}_\sim$ of equivalence classes of the relation $\sim$ satisfies

$$\mathsf{ind}_\sim \leq P \cdot (\mathsf{tower}\,(n, k+1))^{\left(|Q| \cdot \prod_{1 \leq i \leq n} 2^{2^{|\phi_i|}}\right)} \leq P \cdot (\mathsf{tower}\,(n, k+1))^{\left(|Q| \cdot 2^{2^{|\varphi|}}\right)}$$

with $|Q|$ the number of states in $\mathcal{G}$. We let $\overline{M} = M + \mathsf{ind}_\sim + 1$. By the pigeon-hole principle, there must exist $M < h < h' \leq \overline{M}$ such that $h \sim h'$.

### 4.1.2 Periodicity property

We define $\Lambda = h' - h$, and now prove that it is a period for $\varphi$ for counter values larger than or equal to $h$. Assume that $\gamma = (\ell, c)$ is a configuration such that $c \geq h$, and define $\gamma' = (\ell, c + \Lambda)$ (note that $c + \Lambda \geq h'$). We show that $\mathcal{G}, \gamma \models \varphi$ if, and only if, $\mathcal{G}, \gamma' \models \varphi$.

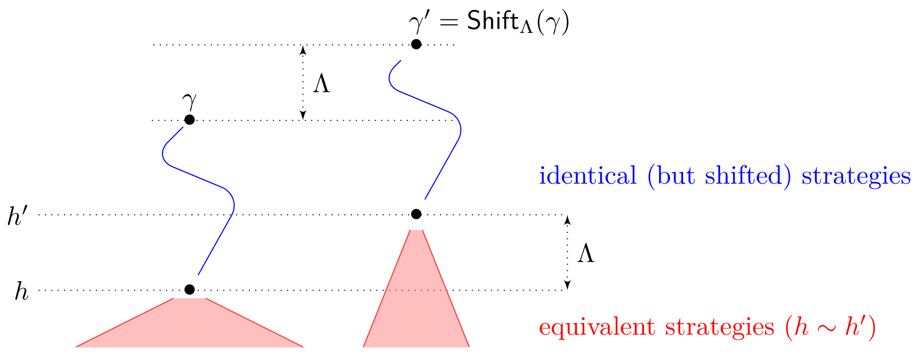



**Figure 5** Construction in Lemma 6 (case (ii)).

▶ Notations. For the rest of this proof, we fix the following notations:

1. if $\rho$ is a run starting with counter value $a > c$, then either the counter always remains above $c$ along $\rho$ (in which case we say that $\rho$ is fully above $c$), or it eventually hits value $c$, and we define $\rho_{\searrow c}$ for the smallest prefix of $\rho$ such that $\mathsf{last}(\rho_{\searrow c})$ has counter value $c$;
2. let $\rho$ be a run that is fully above $M$, and let $c$ be the least counter value appearing in $\rho$. For every $\nu \geq M - c$, we write $\mathsf{Shift}_\nu(\rho)$ for the run $\rho'$ obtained from $\rho$ by shifting the counter value by $\nu$. It is a real run since the counter values along $\rho'$ are also all above $M$.
3. if $D$ is a tuple of states of the deterministic automata $\mathcal{D}_i$, and if $\rho$ is a finite run of $\mathcal{G}$ that is fully above $M$, then we write $D_{+\rho}$ for the image of $D$ after reading $\rho$.

Let $0 \leq j \leq k$. We assume that $\chi_j$ and $\chi'_j$ are two contexts for $\{x_1, \ldots, x_j\}$, and $D$ is a tuple of states of the $\mathcal{D}_i$'s. We write $\mathbb{R}^{D,j}_{(\gamma,\gamma')}(\chi_j, \chi'_j)$ if the following property holds for any run $\rho$ from $\gamma$:

**(i)** if $\rho$ is fully above $h$ (or equivalently, if $\rho' = \mathsf{Shift}_{+\Lambda}(\rho)$, which starts from $\gamma'$, is fully above $h'$), then for every $1 \leq g \leq j$, $\chi_j(x_g)(\rho) = \chi'_j(x_g)(\rho')$;

**(ii)** if $\rho$ is not fully above $h$ (equivalently, if $\rho' = \mathsf{Shift}_{+\Lambda}(\rho)$ is not fully above $h'$), then we decompose $\rho$ (resp. $\rho'$) w.r.t. $h$ (resp. $h'$) and write $\rho = \rho_{\searrow h} \cdot \overline{\rho}$ and $\rho' = \rho'_{\searrow h'} \cdot \overline{\rho}'$. Then:

$$\mathsf{Id}_{\chi_{j\overrightarrow{\rho_{\searrow h}}}}(\mathsf{last}(\rho_{\searrow h}), \widetilde{D}) = \mathsf{Id}_{\chi'_{j\overrightarrow{\rho'_{\searrow h'}}}}(\mathsf{last}(\rho'_{\searrow h'}), \widetilde{D})$$

with $\widetilde{D} = D_{+\rho_{\searrow h}} = D_{+\rho'_{\searrow h'}}$. Recall that $\chi_{j\overrightarrow{\rho_{\searrow h}}}$ shifts all strategies in context $\chi_j$ after the prefix $\rho_{\searrow h}$ (that is, $\chi_j$ is the strategy such that $\chi_{j\overrightarrow{\rho_{\searrow h}}}(\pi) = \chi_j(\rho_{\searrow h} \cdot \pi)$ for every $\pi$).

We then have:

▶ **Lemma 6.** *Fix $0 \leq j < k$, and assume that $\mathbb{R}^{D,j}_{(\gamma,\gamma')}(\chi_j, \chi'_j)$ holds true. Then:*

1. *for every strategy $v$ for $x_{j+1}$ from $\gamma$, one can build a strategy $\mathcal{T}(v)$ for $x_{j+1}$ from $\gamma'$ such that $\mathbb{R}^{D,j+1}_{(\gamma,\gamma')}(\chi_j \cup \{v\}, \chi'_j \cup \{\mathcal{T}(v)\})$ holds true;*
2. *for every strategy $v'$ for $x_{j+1}$ from $\gamma'$, one can build a strategy $\mathcal{T}^{-1}(v')$ for $x_{j+1}$ from $\gamma$ such that $\mathbb{R}^{D,j+1}_{(\gamma,\gamma')}(\chi_j \cup \{\mathcal{T}^{-1}(v')\}, \chi'_j \cup \{v'\})$ holds true.*

**Sketch of proof.** The idea is the following: either we are in case (1), in which case identical (but shifted) strategies can be applied; or we are in case (2), in which case identical (but shifted) strategies can be applied until counter value $h$ (resp. $h'$) is hit, in which case equality of identifiers allows to apply equivalent strategies. The construction is illustrated in Fig. 5. ◀

We use this lemma to transfer a proof that $\gamma \models_\emptyset \varphi$ to a proof that $\gamma' \models_\emptyset \varphi$. We decompose the proof of this equivalence into two lemmas:

▶ **Lemma 7.** *Fix $D^0$ for the tuple of initial states of the $\mathcal{D}_i$'s. Assume that $\mathbb{R}^{D^0,k}_{(\gamma,\gamma')}(\chi,\chi')$ holds (for full contexts $\chi$ and $\chi'$). Let $1 \le i \le n$, and write $\rho = \mathsf{Out}(\gamma,\beta_i[\chi])$ and $\rho' = \mathsf{Out}(\gamma',\beta_i[\chi'])$. Then $\rho \models \phi_i$ if and only if $\rho' \models \phi_i$. In particular, $\gamma \models_\chi f((\beta_i\phi_i)_{1\le i\le n})$ if and only if $\gamma' \models_{\chi'} f((\beta_i\phi_i)_{1\le i\le n})$.*

**Sketch of proof.** As long as runs are above $h$ (resp. $h'$) they visit states that satisfy exactly the same atomic properties (atomic propositions and counter constraints), hence they progress in each $\mathcal{D}_i$ along the same run. When value $h$ (resp. $h'$) is hit, they are generated by strategies that have the same level-0 id, which precisely means they are equivalently accepted by each $\mathcal{D}_i$. Hence both outcomes satisfy the same formulas $\phi_i$ under binding $\beta_i[\chi]$ (resp. $\beta_i[\chi']$). ◀

We finally show the following lemma, by induction on the context, and by noticing that $h \sim h'$ precisely implies the induction property at level 0.

▶ **Lemma 8.** *$\gamma \models_\emptyset \varphi$ if and only if $\gamma' \models_\emptyset \varphi$.*

This allows to conclude with the following corollary:

▶ **Corollary 9.** *$\Lambda$ is a period for the satisfiability of $\varphi$ for configurations with counter values larger than or equal to $h$.*

*Furthermore, $h + \Lambda$ is bounded by $M + P \cdot (\mathsf{tower}\,(n, k+1))^{|Q|\cdot\prod_{1\le i\le} 2^{2^{|\varphi|}}} + 1$.*

▶ Remark. Note that the above proof of existence of a period, though effective (a period can be computed by computing the truth of identifier predicates), does not allow for an algorithm to decide the model-checking problem. One possible idea to lift that periodicity result to an effective algorithm would be to bound the counter values; however things are not so easy: in Fig. 5, equivalent strategies from $h$ and $h'$ might generate runs with (later on) counter values larger than $h$ or $h'$. The decidability status of $\mathsf{1cSL}^1[\mathsf{BG}]$ (and of $\mathsf{1cSL}[\mathsf{BG}]$) model checking remains open.

## 4.2 Extension to 1cSL[BG]

We explain how we can extend the previous periodicity analysis to the full logic $\mathsf{1cSL}[\mathsf{BG}]$. We fix a formula of $\mathsf{1cSL}^{k+1}[\mathsf{BG}]$

$$\varphi = Q_1x_1 \dots Q_kx_k \cdot f((\beta_i\phi_i)_{1\le i\le n})$$

with the same notations than the ones at the beginning of the previous subsection, but $\phi_i$ can use closed formulas of $\mathsf{1cSL}^k[\mathsf{BG}]$ as subformulas.

Let $\Psi_\varphi$ be the set of closed subformulas of $\mathsf{1cSL}^k[\mathsf{BG}]$ that appear directly under the scope of some $\phi_i$. We will replace subformulas of $\Psi_\varphi$ by other formulas involving only (new) atomic propositions and counter constraints. Pick $\psi \in \Psi_\varphi$. Let $h_\psi$ and $\Lambda_\psi$ be the threshold and the period mentioned in Corollary 9. For every location $\ell$ of the game, the set of counter values $c$ such that $(\ell, c) \models \psi$ can be written as $S^\psi_\ell$ (we use a non-periodic set for the values smaller than $h_\psi$ and a periodic set of period $\Lambda_\psi$ for the values above $h_\psi$)—note that we know such a set exists, even though there is (for now) no effective procedure to express it. The size of formula $S^\psi_\ell$ is 1 (we do not take into account the complexity of writing the precise sets used in the constraint). Expand the set of atomic propositions $\mathsf{AP}$ with an extra atomic proposition for each location, say $p_\ell$ for location $\ell$, which holds only at location $\ell$. For every $\psi \in \Psi_\varphi$, replace that occurrence of $\psi$ in $\varphi$ by formula $\bigwedge_{\ell\in L} p_\ell \to (\mathsf{cnt} \in S^\psi_\ell)$. This defines formula $\varphi'$, which is now a $\mathsf{1cSL}^0[\mathsf{BG}]$ formula, and holds equivalently (w.r.t. $\varphi$) from every configuration of $\mathcal{G}$. The size of $\varphi'$ is that of $\varphi$. We apply the result of the previous subsection and get a proof of periodicity of the satisfaction relation for $\varphi'$, hence for $\varphi$.

It remains to compute bounds on the overall period $\Lambda_\varphi$ and threshold $h_\varphi$. The modulo constraints in $\varphi'$ involve periods $\Lambda_\psi$ ($\psi \in \Psi_\varphi$), and the constants used are bounded by $h_\psi$. So the bound $M_{\varphi'}$ is bounded by $\max(\max_{\psi\in\Psi}(h_\psi), M_\varphi)$ where $M_\varphi$ is the maximal constant used in $\varphi$, and the value $P_{\varphi'}$ is the l.c.m. of the periods used in $\varphi$ (call it $P_\varphi$) and of the $\Lambda_\psi$'s (for $\psi \in \Psi_\varphi$): hence $P_{\varphi'} \leq P_\varphi \cdot \max_{\psi\in\Psi_\varphi}(\Lambda_\psi)^{|\varphi|}$ Hence for formula $\varphi'$, we get

$$h_{\varphi'} + \Lambda_{\varphi'} \quad \leq \quad M_{\varphi'} + P_{\varphi'} \cdot \mathsf{tower}\,(n_\varphi, k_\varphi + 1)^{|Q|\cdot 2^{2^{|\varphi'|}}} + 1$$

We infer the following order of magnitude for $h_\varphi + \Lambda_\varphi$, where $\omega_{\Psi_\varphi} = \max_{\psi\in\Psi_\varphi} \omega_\psi$:

$$\begin{aligned}\omega_\varphi &\approx \omega_{\Psi_\varphi} + M_\varphi^{|\varphi|} \cdot (\max \Lambda_\psi)^{|\varphi|} \cdot \mathsf{tower}\,(n_\varphi, k_\varphi + 1)^{|Q|\cdot 2^{2^{|\varphi|}}} \\ &\approx M_\varphi^{|\varphi|} \cdot \omega_{\Psi_\varphi}^{|\varphi|} \cdot \mathsf{tower}\,(n_\varphi, k_\varphi + 1)^{|Q|\cdot 2^{2^{|\varphi|}}}\end{aligned}$$

Using notations of Theorem 5, the order of magnitude can therefore be bounded by

$$\mathsf{tower}\left(\max_{\theta\in\mathsf{Subf}(\varphi)} n_\theta, \max_{\theta\in\mathsf{Subf}(\varphi)} k_\theta + 1\right)^{|Q|\cdot 2^{2^{|\varphi|}}}.$$

▶ Remark. Note that this proof is non-constructive, even for the period and the threshold, since it relies on the model-checking of subformulas, which we don't know how to do. We can nevertheless effectively compute a threshold and a period by taking the l.c.m. of all the integers up to the bound over the period and threshold given in this proof.

## 5 Conclusion

In this paper, we investigated a quantitative extension of Strategy Logic (and more precisely, of its *Boolean-Goal* fragment) over games played on one-counter games. We proved that the corresponding model-checking problem enjoys a nice periodicity property, which we see as a first step towards proving decidability of the problem. We proved however that, if decidable, the problem is hard; this is proved by showing that model checking the fragment SL[BG] over finite-state games is Tower-hard, hence answering an open question from [32].

We are now trying to see how our periodicity property can be used to prove decidability of the model-checking problem. While such a periodicity property helps getting effective algorithms for model checking CTL over one-counter machines [28], the game setting used here makes things much harder. Other further works also include the more general logic 1cSL, whose decidability status (and complexity) is also open. Finally, we did not manage to extend our hardness proof to turn-based games. It would be nice to understand whether the restriction to turn-based games would make 1cSL[BG] (and SL[BG]) model checking easier.

## References


1. R. Alur and D. L. Dill. Automata for modeling real-time systems. In ICALP'90, LNCS 443, pp. 322–335. Springer, 1990.
2. R. Alur and T. A. Henzinger. A really temporal logic. *J. of the ACM*, 41(1):181–203, 1994.
3. R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J. of the ACM*, 49(5):672–713, 2002.
4. K. Apt and E. Grädel. *Lectures in Game Theory for Computer Scientists*. Cambridge University Press, 2011.

5 E. Asarin, O. Maler, A. Pnueli, and J. Sifakis. Controller synthesis for timed automata. In SSC'98, pp. 469–474. Elsevier Science, 1998.
6 A. Bohy, V. Bruyère, E. Filiot, and J.-F. Raskin. Synthesis from LTL specifications with mean-payoff objectives. In TACAS'13, LNCS 7795, pp. 169–184. Springer, 2013.
7 U. Boker, K. Chatterjee, T. A. Henzinger, and O. Kupferman. Temporal specifications with accumulative values. *ACM Transactions on Computational Logic*, 15(4):27:1–27:25, 2014.
8 P. Bouyer, U. Fahrenberg, K. G. Larsen, N. Markey, and J. Srba. Infinite runs in weighted timed automata with energy constraints. In FORMATS'08, LNCS 5215, pp. 33–47. Springer, 2008.
9 P. Bouyer, P. Gardy, and N. Markey. Quantitative verification of weighted Kripke structures. In ATVA'14, LNCS 8837, pp. 64–80. Springer, 2014.
10 P. Bouyer, P. Gardy, and N. Markey. Weighted strategy logic with boolean goals over one-counter games. Research Report LSV-15-08, Laboratoire Spécification et Vérification, ENS Cachan, France, 2015. 26 pages.
11 T. Brázdil, P. Jančar, and A. Kučera. Reachability games on extended vector addition systems with states. In ICALP'10, LNCS 6199, pp. 478–489. Springer, 2010.
12 S. V. A. Campos and E. M. Clarke. Real-time symbolic model checking for discrete time models. In *Real-time symbolic model checking for discrete time models*, AMAST Series in Computing 2, pp. 129–145. World Scientific, 1995.
13 P. Čermák, A. Lomuscio, and A. Murano. Verifying and synthesising multi-agent systems against one-goal strategy logic specifications. In AAAI'15, pp. 2038–2044. AAAI Press, 2015.
14 A. Chakrabarti, L. de Alfaro, T. A. Henzinger, and M. Stoelinga. Resource interfaces. In EMSOFT'03, LNCS 2855, pp. 117–133. Springer, 2003.
15 K. Chatterjee, L. Doyen, T. A. Henzinger, and J.-F. Raskin. Generalized mean-payoff and energy games. In FSTTCS'10, LIPIcs 8, pp. 505–516. Leibniz-Zentrum für Informatik, 2010.
16 K. Chatterjee, T. A. Henzinger, and N. Piterman. Strategy logic. In CONCUR'07, LNCS 4703, pp. 59–73. Springer, 2007.
17 K. Chatterjee, M. Randour, and J.-F. Raskin. Strategy synthesis for multi-dimensional quantitative objectives. Research Report 1201.5073, arXiv, 2012.
18 E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In LOP'81, LNCS 131, pp. 52–71. Springer, 1982.
19 E. M. Clarke, O. Grumberg, and D. A. Peled. *Model checking.* MIT Press, 2000.
20 A. Da Costa, F. Laroussinie, and N. Markey. ATL with strategy contexts: Expressiveness and model checking. In FSTTCS'10, LIPIcs 8, pp. 120–132. Leibniz-Zentrum für Informatik, 2010.
21 L. de Alfaro, M. Faella, T. A. Henzinger, R. Majumdar, and M. Stoelinga. The element of surprise in timed games. In CONCUR'03, LNCS 2761, pp. 142–156. Springer, 2003.
22 S. Demri and R. Gascon. The effects of bounding syntactic resources on Presburger LTL. *Journal of Logic and Computation*, 19(6):1541–1575, 2009.
23 A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8(2):109–113, 1979.
24 E. A. Emerson, A. K.-L. Mok, A. P. Sistla, and J. Srinivasan. Quantitative temporal reasoning. *Real-Time Systems*, 4:331–352, 1992.
25 J. Esparza and M. Nielsen. Decidability issues for Petri nets – a survey. *EATCS Bulletin*, 52:244–262, 1994.
26 U. Fahrenberg, L. Juhl, K. G. Larsen, and J. Srba. Energy games in multiweighted automata. In ICTAC'11, LNCS 6916, pp. 95–115. Springer, 2011.

27 S. Göller, C. Haase, J. Ouaknine, and J. Worrell. Model checking succinct and parametric one-counter automata. In ICALP'10, LNCS 6199, pp. 575–586. Springer, 2010.

28 S. Göller and M. Lohrey. Branching-time model checking of one-counter processes. In STACS'10, LIPIcs 20, pp. 405–416. Leibniz-Zentrum für Informatik, 2010.

29 R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.

30 F. Laroussinie and N. Markey. Augmenting ATL with strategy contexts. *Inf. & Comp.*, 2015. To appear.

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

32 F. Mogavero, A. Murano, G. Perelli, and M. Y. Vardi. Reasoning about strategies: On the model-checking problem. *ACM Transactions on Computational Logic*, 15(4):34:1–34:47, 2014.

33 F. Mogavero, A. Murano, and L. Sauro. A behavioral hierarchy of strategy logic. In CLIMA'14, LNAI 8624, pp. 148–165. Springer, 2014.

34 F. Mogavero, A. Murano, and M. Y. Vardi. Reasoning about strategies. In FSTTCS'10, LIPIcs 8, pp. 133–144. Leibniz-Zentrum für Informatik, 2010.

35 A. Pnueli. The temporal logic of programs. In FOCS'77, pp. 46–57. IEEE Comp. Soc. Press, 1977.

36 J.-P. Queille and J. Sifakis. Specification and verification of concurrent systems in CESAR. In SOP'82, LNCS 137, pp. 337–351. Springer, 1982.

37 J. Reichert. On the complexity of counter reachability games. In RP'13, pp. 196–208. Abdulla, Parosh Aziz and Potapov, Igor, 2013.

38 S. Schmitz. Complexity hierarchies beyond elementary. Research Report cs.CC/1312.5686, arXiv, 2013.

39 O. Serre. Parity games played on transition graphs of one-counter processes. In FoSSaCS'06, LNCS 3921, pp. 337–351. Springer, 2006.

40 A. P. Sistla. *Theoretical Issues in the Design and Verification of Distributed Systems*. PhD thesis, Harvard University, Cambridge, Massachussets, USA, 1983. ?

41 A. P. Sistla, M. Y. Vardi, and P. Wolper. The complementation problem for Büchi automata, with applications to temporal logic. In ICALP'85, LNCS 194, pp. 465–474. Springer, 1985.

42 W. Thomas. Infinite games and verification (extended abstract of a tutoral). In CAV'02, LNCS 2404, pp. 58–64. Springer, 2002.

43 S. Vester. On the complexity of model-checking branching and alternating-time temporal logics in one-counter systems. In ATVA'15, LNCS, Lecture Notes in Computer Science. Springer, 2015.

44 U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theor. Computer Science*, 158(1-2):343–359, 1996.