On the semantics of Strategy Logic[☆]

Patricia Bouyer, Patrick Gardy, Nicolas Markey

LSV – CNRS, ENS Cachan, Univ. Paris-Saclay – France

Abstract

We define and study a slight variation on the semantics of Strategy Logic: while in the classical semantics, all strategies are shifted during the evaluation of temporal modalities, we propose to only shift the strategies that have been assigned to a player, thus matching the intuition that we can assign the *very same* strategy to the players at different points in time. We prove that surprisingly, this renders the model-checking problem undecidable.

Keywords: Multi-agent systems; Strategic reasoning; Temporal logics.

1. Introduction

Model checking [CGP00, BK08] is a model-based technique for automatically verifying properties of computerized systems. Model-checking algorithms exhaustively explore the set of behaviours of the (model of the) system under study, and compare this set against properties being checked. *Temporal logics*, and in particular the *Linear-time Temporal Logic* (LTL) [Pnu77] and the *Computation-Tree Logic* (CTL) [QS82, CE82], provide a convenient formalism for expressing such properties: they extend boolean logics in order to state properties of sequences of boolean valuations. Using temporal modalities, they can constrain the order in which various events occur along such sequences. It is then possible to express, for instance, that any problem is eventually followed by an alarm.

During the last 15 years, temporal logics—and model checking—have been extended to deal with *multi-agent systems*: there, the behaviour of the global system depends on the actions of individual agents, and the new logic, *Alternating-time Temporal Logic* (ATL) [AHK98, AHK02], can now express what some agent can (or cannot) achieve, or what happens in the whole system when some agent tries to achieve their goal. This extension is particularly relevant in the setting of *controller synthesis*, as it provides a way of expressing the existence of a

[☆]This work was partly supported by ERC project EQualIS (FP7-308087) and FET project Cassting (FP7-601148).

Email addresses: patricia.bouyer@lsv.fr (Patricia Bouyer), patrick.gardy@lsv.fr (Patrick Gardy), nicolas.markey@lsv.fr (Nicolas Markey)

controller (often seen as a *strategy* in a game against the other agents) enforcing a given property.

However, it has been noticed recently that ATL is not expressive enough to express many interesting properties of multi-agent systems. In particular, ATL is mainly usable for expressing properties of antagonistic agents, and cannot express real interactions or collaborations between agents. It has thus been enriched in order to allow for such collaborations: *Strategy Logic* (SL) [CHP07, CHP10, MMV10, MMPV14], in particular, deals with strategies as first-class citizens, with (first-order) quantification, and assignment to one or several agents.

Consider for instance a network of several clients, that may ask a central server for accessing a shared resource. One (or several) users can turn the clients on and off, and when turned on, each client then requests access to the resource. The server then has two objectives: one is to enforce that no two clients access the resource at the same time, whatever the clients do; the second property is that the clients must have a strategy that each of them can apply when turned on, and that ensures access to the resource (by collaborating with the server). This, in SL (with adapted syntax to make the formula readable), would be written

$$\exists \sigma_{\mathsf{server}}.\ \texttt{if server applies } \sigma_{\mathsf{server}} \texttt{ then } \Big[(\texttt{always mutual exclusion}) \wedge (\exists \sigma_{\mathsf{client}}.\ \mathsf{always}\ (\texttt{if client applies } \sigma_{\mathsf{client}} \texttt{ then eventually access}))\Big].$$

SL model checking is decidable [CHP07, MMPV14]. In this paper, we prove that this result heavily relies on a semantical choice that is silently made in the previous papers about SL. We argue in this paper that this semantical choice does *not* achieve the expected meaning for the sample formula above (intuitively, because it gives to the subformula "client applies σ_{client}" a meaning that depends on the history of the system, whereas when a client is turned on, it should start applying its strategy with no prior knowledge about what has happened previously). We propose an alternative semantics, which assumes that strategies starts being applied with empty history, and prove that this minor change makes the model-checking problem undecidable.

2. Definitions

2.1. Turn-based games

Logics for multi-agent systems are usually interpreted over structures involving multiple agents (hence the name...). In the context of this note, we only focus on *two-player turn-based* games, since this is enough for proving our result.

Definition 1. *A two-player turn-based game is a tuple $\mathcal{G} = \langle S_{\bigcirc}, S_{\Box}, T\rangle$ where $S_{\bigcirc}$ and $S_{\Box}$ are pairwise-disjoint finite sets of states, $T \subseteq S^2$ (where $S = S_{\bigcirc} \cup S_{\Box}$) is the set of transitions. It is assumed that for all $s \in S$, there exists $s' \in S$ s.t. $(s, s') \in T$.*

A path in such a game is a (finite or infinite) sequence $(s_i)_{1\leq i<L+1}$ (with $L \in \mathbb{N} \cup \{+\infty\}$) of states such that, for every $1 \leq i < L$, it holds $(s_i, s_{i+1}) \in T$. The length of a path $(s_i)_{1\leq i<L+1}$ is the number L of elements of the sequence. A strategy for Player $\bigcirc$ is a mapping $\sigma^{\bigcirc}\colon S^* \times S_{\bigcirc} \to S$ such that for all finite path $(s_i)_{1\leq i\leq n}$ with $s_n \in S_{\bigcirc}$, it holds $(s_n, \sigma^{\bigcirc}((s_i)_{1\leq i\leq n})) \in T$. In other terms, a strategy for Player $\bigcirc$ tells which transition to follow after any finite play ending in a state controlled by that player. Strategies for Player $\Box$ are defined symmetrically. We write $\mathsf{Strat}^{\bigcirc}$ and $\mathsf{Strat}^{\Box}$ for the sets of strategies of Players $\bigcirc$ and $\Box$, and Strat for the set of all strategies.

Given a strategy $\sigma^{\bigcirc}$ for Player $\bigcirc$, a strategy $\sigma^{\Box}$ for Player $\Box$, and a state s, the outcome of $\sigma^{\bigcirc}$ and $\sigma^{\Box}$ from s is the infinite path $(s_i)_{i\geq 1}$ s.t. $s_1 = s$, and $s_{n+1} = \sigma^{\bigcirc}((s_i)_{1\leq i\leq n})$ if $s_n \in S_{\bigcirc}$, and $s_{n+1} = \sigma^{\Box}((s_i)_{1\leq i\leq n})$ if $s_n \in S_{\Box}$.

2.2. Strategy Logic (SL*)*

2.2.1. Syntax and semantics of SL

We now present logics for expressing properties of the games defined above. For this, we first fix a finite set AP of atomic propositions, and consider *labelled* games, with a mapping $\ell\colon S \to 2^{\mathsf{AP}}$.

Strategy Logic (SL for short) was introduced in [CHP07], and further extended and studied in [MMV10, MMPV14], as a rich logical formalism for expressing properties of games. Formulas in SL are built along the following grammar[1]:

$$\mathsf{SL} \ni \varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi \mid \exists x.\ \varphi \mid \mathsf{assign}(a \mapsto x).\ \varphi$$

where p ranges over AP, x ranges over a set Var of variables, and a ranges over a finite set Agt of agents (in our setting, $\mathsf{Agt} = \{\bigcirc, \Box\}$). Thus, SL can be seen as an extension of LTL [Pnu77] with strategy quantification ($\exists x.\ \varphi$, which selects a strategy and stores it in variable x, before evaluating φ) and strategy assignments ($\mathsf{assign}(a \mapsto x).\ \varphi$, which assigns the strategy stored in variable x to Player a, and then evaluates φ).

Formally, formulas of SL are evaluated at a state s of a game $\mathcal{G}$, under a valuation χ mapping (part of the) agents and variables to strategies. We write $\mathsf{dom}(\chi)$ for the subset of $\mathsf{Agt} \cup \mathsf{Var}$ on which χ is defined. The semantics of atomic propositions and boolean combinators is the natural one.

In order to define the semantics of strategy quantifiers and assignments, we need several intermediary notions. The set of *free agents and variables* of a formula φ, which we write $\mathsf{free}(\varphi)$, contains the agents and variables that have to be associated with a strategy before φ can be evaluated. It is defined inductively

[1] We mainly follow the syntax of SL from [MMPV14], but write $\exists x.\ \varphi$ instead of $\langle\langle x\rangle\rangle\,\varphi$ (to avoid confusions with the ATL strategy quantified $\langle\langle\,\text{-}\,\rangle\rangle$), and $\mathsf{assign}(a \mapsto x).\ \varphi$ instead of $(a, x)\varphi$ (thus avoiding overloading parentheses).

as follows:

$$\mathsf{free}(p) = \varnothing \quad \text{for all } p \in \mathsf{AP} \qquad \mathsf{free}(\mathbf{X}\,\varphi) = \mathsf{Agt} \cup \mathsf{free}(\varphi)$$
$$\mathsf{free}(\neg\,\varphi) = \mathsf{free}(\varphi) \qquad \mathsf{free}(\varphi\,\mathbf{U}\,\psi) = \mathsf{Agt} \cup \mathsf{free}(\varphi) \cup \mathsf{free}(\psi)$$
$$\mathsf{free}(\varphi \vee \psi) = \mathsf{free}(\varphi) \cup \mathsf{free}(\psi) \qquad \mathsf{free}(\exists x.\ \varphi) = \mathsf{free}(\varphi) \setminus \{x\}$$
$$\mathsf{free}(\mathsf{assign}(a \mapsto x).\ \varphi) = \begin{cases} \mathsf{free}(\varphi) & \text{if } a \notin \mathsf{free}(\varphi) \\ (\mathsf{free}(\varphi) \cup \{x\}) \setminus \{a\} & \text{otherwise} \end{cases}$$

Let s be a state of $\mathcal{G}$, χ be a valuation, $x \in \mathsf{Var}$, and $\varphi \in \mathsf{SL}$ s.t. $\mathsf{free}(\varphi)\setminus\{x\} \subseteq \mathsf{dom}(\chi)$. Then

$$\mathcal{G}, s \models_\chi \exists x.\varphi \quad \Leftrightarrow \quad \exists \sigma \in \mathsf{Strat}.\ \mathcal{G}, s \models_{\chi[x \mapsto \sigma]} \varphi.$$

If additionally $a \in \mathsf{Agt}$ is such that $(\mathsf{free}(\varphi) \setminus \{a\}) \cup \{x\} \subseteq \mathsf{dom}(\chi)$, then

$$\mathcal{G}, s \models_\chi \mathsf{assign}(a \mapsto x).\ \varphi \quad \Leftrightarrow \quad \mathcal{G}, s \models_{\chi[a \mapsto \chi(x)]} \varphi.$$

Notice that it could be the case that $\chi(a)$ is already defined; then the previous value of $\chi(a)$ is discarded.

Remark 2. *In many cases, strategies are assigned immediately after being selected, and it will be convenient to have a shorthand for this: we write $\langle\!\langle \cdot a \cdot \rangle\!\rangle\,\varphi$ for the formula $\exists x.\ \mathsf{assign}(a \mapsto x).\ \varphi$, where x is a variable that does not occur in φ. The dual construct $[\![\cdot a \cdot]\!]\,\varphi = \neg\,\langle\!\langle a \rangle\!\rangle\,\neg\,\varphi$ will also be useful: it states that any strategy of Player a (added to the current context) has at least one outcome where φ holds.*

Notice that the logic obtained from SL by allowing only $\langle\!\langle \cdot a \cdot \rangle\!\rangle\,\varphi$ (in place of strategy quantification $\exists x.\varphi$ and strategy assignment $\mathsf{assign}(a \mapsto x).\ \varphi$) is precisely the logic ATL^*_{sc} of [DLM10, LM15].

It remains to define the semantics of temporal modalities $\mathbf{X}$ and $\mathbf{U}$. For this, we need to *shift* strategy: given a strategy σ and a finite path $\pi = (s_i)_{1 \le i \le n}$, the shifted strategy $\sigma_{\overrightarrow{\pi}}$ is defined as

$$\sigma_{\overrightarrow{\pi}}(\rho) = \sigma(\pi \cdot \rho)$$

for all finite paths ρ that start in the last state of π. For a valuation χ, we define $\chi_{\overrightarrow{\pi}}$ as the valuation obtained from χ by shifting all strategies in the image of χ by π. If $\mathsf{Agt} \subseteq \mathsf{dom}(\chi)$, then from a given state s and a given integer n, χ induces a unique outcome of length n from s, which we write $\mathsf{out}_n(s, \chi(\mathsf{Agt}))$. We then define $\chi_{\overrightarrow{n}}$ as the valuation obtained by shifting all strategies in the image of χ by $\mathsf{out}_n(s, \chi(\mathsf{Agt}))$. Under the same conditions, we also define $s^{\chi}_{\overrightarrow{n}}$ as the last state of $\mathsf{out}_n(s, \chi(\mathsf{Agt}))$.

Now, if $\mathsf{Agt} \cup \mathsf{free}(\varphi) \cup \mathsf{free}(\psi) \subseteq \mathsf{dom}(\chi)$

$$\begin{aligned} \mathcal{G}, s \models_\chi \mathbf{X}\,\varphi \quad &\Leftrightarrow \quad \mathcal{G}, s^{\chi}_{\overrightarrow{1}} \models_{\chi_{\overrightarrow{1}}} \varphi \\ \mathcal{G}, s \models_\chi \psi\,\mathbf{U}\,\varphi \quad &\Leftrightarrow \quad \exists k \in \mathbb{N}.\ \mathcal{G}, s^{\chi}_{\overrightarrow{k}} \models_{\chi_{\overrightarrow{k}}} \varphi \text{ and} \\ &\qquad\qquad \forall j \in \mathbb{N}.\ [0 \le j < k \ \Rightarrow\ \mathcal{G}, s^{\chi}_{\overrightarrow{j}} \models_{\chi_{\overrightarrow{j}}} \psi]. \end{aligned}$$

2.2.2. An alternative semantics for SL

The above definition corresponds to the semantics of [MMV10, MMPV14]. The seminal papers [CHP07, CHP10] about SL do not formally define the semantics of the temporal modalities, simply saying that "*[t]he semantics of path formulas is the usual semantics of* LTL". Our modified definition below precisely changes the definition of the LTL modalities, while still sticking to their *usual semantics*.

One important thing to notice about the semantics above is that all strategies are shifted during the evaluation of temporal modalities, independently of whether they are currently being played by a player or not. A natural alternative would be to only shift those strategies that the agents are playing, and to freeze the strategies that are stored in variables. Indeed, a nice feature of SL is that a single strategy can be assigned to the same or different[2] players at different points in time. This raises the semantic question about what is meant by a *single* strategy, when considered after different histories: should the strategy start with empty history, or should it take into account the different histories? To the best of our knowledge, only the latter definition has been considered in the literature.

We believe that the former approach is relevant in some situations, and thus propose a slight change in the definition of the semantics of SL: we write $\mathsf{SL}^{\mathsf{alt}}$ for the logic obtained from SL by only changing the way a valuation χ (with $\mathsf{Agt} \subseteq \mathsf{dom}(\chi)$) are shifted, now using the following definition:

$$\begin{array}{rcl} \chi_{\overrightarrow{\pi}} : s \in \mathsf{dom}(\chi) \cap \mathsf{Var} & \mapsto & \chi(s) \\ a \in \mathsf{Agt} & \mapsto & \chi(a)_{\overrightarrow{\pi}} \end{array}$$

Under this definition, only the strategies of the players are effectively shifted, while the strategies stored in the variables are kept unchanged.

Example 3. *The client-server example developed in the introduction is a case where strategies should remain "idle" as long as they are not played: the intention is indeed to always play the same sequence of actions when starting from the same configuration, instead of having this sequence depend on the whole history of the system.*

Remark 4. *Notice that both semantics coincide when considering only memoryless strategies. This is also trivially the case when strategies are assigned immediately after being quantified, as is the case in* ATL^*_{sc} *[DLM10], and more generally, when no temporal modality is allowed between strategy quantification and strategy assignment, as in* $\mathsf{SL}[\text{BG}]$ *[MMPV14].*

3. Undecidability proof

Our main result in this paper is the following theorem:

[2]Notice that this is not relevant in our setting of turn-based games, as a Player-◯ strategy cannot be played by Player □. But this could make sense under specific circumstances in concurrent games.

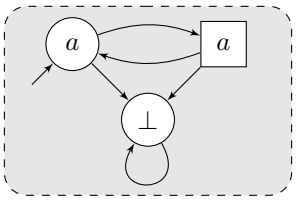

Fig. 1: Module counter(a)

Theorem 5. *Model checking* $\mathsf{SL}^{\mathsf{alt}}$ *is undecidable (over two-player turn-based games, hence also over richer models of games).*

Remark 6. *Before entering the technical proof, let us first comment on the model-checking algorithm for* SL*, and why it fails for* $\mathsf{SL}^{\mathsf{alt}}$*. First notice that a strategy for some player(s) is nothing but a labelling of the computation tree of the game with moves of the considered player(s). The properties that such a labelling has to satisfy can then be expressed using* CTL^* *over such a labelled computation tree. This reduces the model-checking problem of* SL *to that of* QCTL*, as explained in [DLM12].*

With our modified semantics, such a translation fails: when selecting a strategy, we would have to keep track of what that strategy would propose depending on the point where it started being played, hence an unbounded labelling.

3.1. Testing equality

Consider the game depicted on Fig. 1. A circle-player strategy in this game can be characterized by a single integer, which tells how many times the strategy moves to $\boxed{a}$ before going to $\textcircled{\bot}$. Similarly for Player $\square$.

Now, given two strategies $\sigma_{\bigcirc}$ and $\sigma_{\square}$, we write an $\mathsf{SL}^{\mathsf{alt}}$ formula to test whether those two strategies correspond to the same integer. The formula reads

$$\mathsf{assign}(\bigcirc \mapsto \sigma_{\bigcirc}, \square \mapsto \sigma_{\square}).\ \varphi_{=}$$

where $\varphi_{=}$ is the formula

$$\mathbf{G}\,(\boxed{a} \Rightarrow \mathbf{X}\,\textcircled{a}) \wedge \mathbf{F}\,\big[(\textcircled{a} \wedge \mathbf{X}\,\textcircled{\bot}) \wedge [\![\cdot\bigcirc\cdot]\!]\,\mathbf{X}\,\mathbf{X}\,\textcircled{\bot}\big] \tag{1}$$

Indeed, assuming that $\sigma_{\bigcirc}$ plays k times to $\boxed{a}$ and then to $\textcircled{\bot}$, then for Formula (1) to hold, it must be the case that Player $\square$ always returns to $\textcircled{a}$ during the first k cycles in the loop; then if instead of going to $\textcircled{\bot}$, Player $\bigcirc$ would go back to $\boxed{a}$, the formula enforces that $\sigma_{\square}^{k}$ would play to $\textcircled{\bot}$, as required.

Using similar ideas, we can characterize the situations where $\sigma_{\bigcirc} = \sigma_{\square} + 1$ (identifying strategies with their associated integer). It suffices to replace $\varphi_{=}$ with the following formula φ_{+1}:

$$\mathbf{G}\,(\textcircled{a} \Rightarrow \mathbf{X}\boxed{a}) \wedge \mathbf{F}\,\Big[(\boxed{a} \wedge \mathbf{X}\,\textcircled{\bot}) \wedge \big[(\,\langle\!\langle\cdot\square\cdot\rangle\!\rangle\,\mathbf{X}\,\mathbf{X}\boxed{a}) \wedge (\,[\![\cdot\square\cdot]\!]\,\mathbf{X}\,\mathbf{X}\,\mathbf{X}\,\mathbf{X}\,\textcircled{\bot})\big]\Big] \tag{2}$$

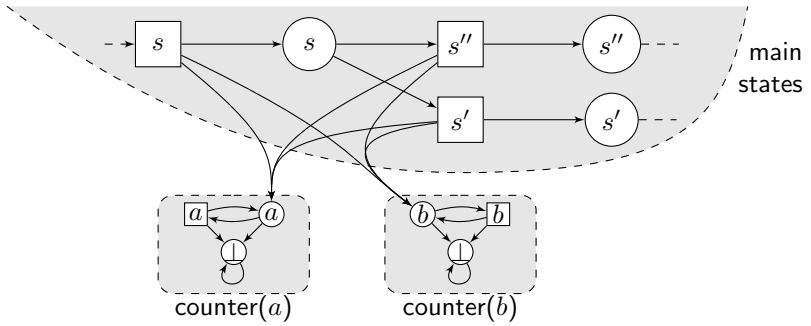

Fig. 2: Schematics representation of the global reduction (for a decrementing state s)

3.2. Proof of Theorem 5

A (deterministic) two-counter machine is a tuple $\mathcal{M} = \langle S, s_0, s_{\mathsf{h}}, \delta \rangle$ where S is a finite set of states, $s_0 \in S$ is the initial state, $s_{\mathsf{h}} \in S$ is the halting state, and $\delta \colon S \to \{c_1, c_2\} \times (S \cup S \times S)$ is the transition relation; transitions of the form $\delta(s) = (c, s')$ increment counter c and go to s', while transitions of the form $\delta(s) = (c, s', s'')$ either go to s' if counter c is zero, or decrement c and go to s'' if $c > 0$. A configuration of $\mathcal{M}$ is a triple $(s, c_1, c_2) \in S \times \mathbb{N} \times \mathbb{N}$, with the initial configuration being $(s_0, 0, 0)$. It is well-known that the reachability of the halting state from the initial configuration is undecidable [Min67].

We encode this problem into a model-checking problem for $\mathsf{SL}^{\mathsf{alt}}$. We begin with describing a two-player turn-based game $\mathcal{G}$ that will be used in this reduction: its states are $(S \times \{\bigcirc, \Box\}) \cup \{\textcircled{a}, \textcircled{b}, \boxed{a}, \boxed{b}, \textcircled{\bot}\}$; in the sequel, we write $\textcircled{s}$ and $\boxed{s}$ to represent states $(s, \bigcirc)$ and $(s, \Box)$, respectively (and require that none of the symbols a, b and $\bot$ belongs to S, to avoid confusion). Those states in $S \times \{\bigcirc, \Box\}$ are called the *main states*, while those in $\{\textcircled{a}, \textcircled{b}, \boxed{a}, \boxed{b}, \textcircled{\bot}\}$ are the *counter states*. The initial state is $\boxed{s_0}$.

Figure 2 depicts the construction: the square-player main states $\boxed{s}$ have three outgoing transitions, going to $\textcircled{s}$, $\textcircled{a}$ and $\textcircled{b}$. The circle-player main states $\textcircled{s}$ have one or two outgoing transitions, depending on the transition from s in $\mathcal{M}$:

- if $\delta(s) = (c, s')$, then $\textcircled{s}$ has a single transition, to $\boxed{s'}$;
- if $\delta(s) = (c, s', s'')$, then $\textcircled{s}$ has two transitions, to $\boxed{s'}$ and $\boxed{s''}$.

In state $\textcircled{a}$, the game is as depicted on Fig. 1. A similar module is used from $\textcircled{b}$ (with states $\textcircled{b}$ and $\boxed{b}$ in place of $\textcircled{a}$ and $\boxed{a}$).

We now explain the intuition behind our construction. The main idea is that, with each strategy of Player $\bigcirc$, we can associate a sequence of configurations of the two-counter machine $\mathcal{M}$. An $\mathsf{SL}^{\mathsf{alt}}$ formula will then express the existence of a circle-player strategy corresponding to a valid execution of $\mathcal{M}$ reaching the halting state.

We define the correspondence between strategies of Player $\bigcirc$ and sequences of configurations of $\mathcal{M}$ as follows. Fix a strategy $\sigma_{\bigcirc}$ of Player $\bigcirc$, and consider

the infinite outcomes of this strategy from $\boxed{s_0}$. One of these outcomes visits only main states: it is made of a sequence of pairs of states of the form $\boxed{s^i} \cdot \textcircled{s^i}$. Hence it defines an (infinite) sequence $(s^i)_{i\in\mathbb{N}}$ of states of $\mathcal{M}$.

At each square state along that infinite outcome (after some finite history h), two additional branches emerge: one going to module $\mathsf{counter}(a)$ via $\textcircled{a}$, and one going to module $\mathsf{counter}(b)$ via $\textcircled{b}$. Following the discussion of Section 3.1, the strategy $\sigma^h_{\bigcirc}$, defined as $\sigma^h_{\bigcirc}(\rho) = \sigma_{\bigcirc}(h \cdot \rho)$ for any path ρ in a $\mathsf{counter}$ module, corresponds to an integer (or $+\infty$). We write c^i_1 for the integer associated with $\sigma^h_{\bigcirc}$ when ρ is a path in $\mathsf{counter}(a)$, and c^i_2 when ρ is a path in $\mathsf{counter}(b)$.

This way, we have associated with $\sigma_{\bigcirc}$ a sequence (s^i, c^i_1, c^i_2) of configurations of $\mathcal{M}$. The rest of the proof consists in building a $\mathsf{SL}^{\mathsf{alt}}$ formula that precisely characterizes the strategy $\sigma_{\bigcirc}$ that corresponds to the (unique) valid run of $\mathcal{M}$ from its initial configuration, and requiring that the *main* outcome reaches s_{h}. Our formula thus looks as follows:

$$\mathsf{assign}(\bigcirc \mapsto \sigma_{\bigcirc}) \left[\varphi_{\mathsf{correct}} \wedge [\![\cdot\Box\cdot]\!]\, \mathbf{F}\left(\boxed{s_{\mathsf{h}}} \vee \textcircled{\perp}\right)\right]. \tag{3}$$

This enforces that the outcome of $\sigma_{\bigcirc}$ that remains in the main states has to visit the target state s_{h}, and that the other outcomes have to end up in $\textcircled{\perp}$, which rules out "infinite counters".

Formula $\varphi_{\mathsf{correct}}$ then has three things to check:

1. that the initial configuration is encoded properly in $\sigma_{\bigcirc}$;
2. that the zero-test transitions are taken according to the current values of the counters;
3. that the counters are incremented or decremented correctly.

Property (1) is enforced by requiring that

$$[\![\cdot\Box\cdot]\!]\ [(\mathbf{X}\textcircled{a} \vee \mathbf{X}\textcircled{b}) \Rightarrow \mathbf{X}\,\mathbf{X}\,\textcircled{\perp}]\,.$$

Indeed, this expresses that if Player $\Box$ leaves the main states at the first step, then Player $\bigcirc$ will directly go to $\textcircled{\perp}$, thus encoding value zero for both counters.

Property (2) requires checking that Player $\bigcirc$ chooses the correct branch in zero-test states, according to the current value of the counters. We can express this as follows:

$$[\![\cdot\Box\cdot]\!]\, \mathbf{G}\left[\bigwedge_{\substack{s \text{ s.t.} \\ \delta(s)=(c,s',s'')}} \boxed{s} \Rightarrow \Big(\left[\langle\!\langle\cdot\Box\cdot\rangle\!\rangle\, (\mathbf{X}\textcircled{c} \Rightarrow \mathbf{X}\,\mathbf{X}\,\textcircled{\perp})\right] \Leftrightarrow \left(\langle\!\langle\cdot\Box\cdot\rangle\!\rangle\, \mathbf{X}\,\mathbf{X}\,\boxed{s'}\right)\Big)\right]$$

where $\textcircled{c}$ is an abuse of notation, meaning $\textcircled{a}$ if $c = c_1$ and $\textcircled{b}$ if $c = c_2$. This formula states that when visiting a state $\boxed{s}$, Player $\Box$ can go to $\textcircled{c}$ and then directly to $\textcircled{\perp}$ (which means that counter c is zero) if, and only if, staying in the main states would take the game to state s' (notice that by construction of $\mathcal{G}$, the only other option is to go to s'').

Property (3) is the most complicated part, and is actually the only point where we will make use of the specific semantics of $\mathsf{SL}^{\mathsf{alt}}$. We reuse here the idea

presented in Section 3.1. For instance, in the case of an incrementing state, we first select a strategy $\sigma_{\Box}^{k}$ that matches the exact number of visits to $\textcircled{a}$ (or $\textcircled{b}$) if Player $\Box$ decides to leave the main states in the present position; such a strategy can be characterized using a formula similar to Formula (1). We then require that, if Player $\Box$ decides to leave the main states at the next step, then playing the same strategy $\sigma_{\Box}^{k}$ (with empty history) would play towards $\textcircled{a}$ (or $\textcircled{b}$) once more than what strategy $\sigma_{\bigcirc}$ proposes. Formally, for the case of a state s with $\delta(s) = (c, s')$, we would write

$$[\![\cdot\Box\cdot]\!]\,\mathbf{G}\,\Big[\boxed{s} \Rightarrow \exists \sigma_{\Box}^{\mathsf{count}}.\ \langle\!\langle\Box\rangle\!\rangle\,\mathbf{X}\,(\textcircled{c} \wedge \mathsf{assign}(\Box \mapsto \sigma_{\Box}^{\mathsf{count}}).\ \varphi_{=}) \wedge \langle\!\langle\Box\rangle\!\rangle\,\mathbf{X}\,\mathbf{X}\,(\boxed{s'} \wedge \mathbf{X}\,(\textcircled{c} \wedge \mathsf{assign}(\Box \mapsto \sigma_{\Box}^{\mathsf{count}}).\ \varphi_{+1}))\Big]$$

Similar formulas can be written for decrementing states (provided the counter is non-zero).

In the end, if there exists a strategy $\sigma_{\bigcirc}$ that makes Formula (3) true in $\boxed{s_0}$, then this strategy corresponds to a halting computation of $\mathcal{M}$. Conversely, a halting computation of $\mathcal{M}$ gives rise to a strategy making Formula (3) hold in $\boxed{s_0}$, which concludes the proof.

References

[AHK98] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In COMPOS'97, LNCS 1536, p. 23–60. Springer, 1998.

[AHK02] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.

[BK08] Ch. Baier and J.-P. Katoen. *Principles of Model-Checking*. MIT Press, 2008.

[CE82] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In LOP'81, LNCS 131, p. 52–71. Springer, 1982.

[CGP00] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model checking*. MIT Press, 2000.

[CHP07] K. Chatterjee, T. A. Henzinger, and N. Piterman. Strategy logic. In CONCUR'07, LNCS 4703, p. 59–73. Springer, 2007.

[CHP10] K. Chatterjee, T. A. Henzinger, and N. Piterman. Strategy logic. *Information and Computation*, 208(6):677–693, 2010.

[DLM10] A. Da Costa, F. Laroussinie, and N. Markey. ATL with strategy contexts: Expressiveness and model checking. In FSTTCS'10, Leibniz International Proceedings in Informatics 8, p. 120–132. Leibniz-Zentrum für Informatik, 2010.

[DLM12] A. Da Costa, F. Laroussinie, and N. Markey. Quantified CTL: Expressiveness and model checking. In CONCUR'12, LNCS 7454, p. 177–192. Springer, 2012.

[LM15] F. Laroussinie and N. Markey. Augmenting ATL with strategy contexts. *Information and Computation*, 2015. To appear.

[Min67] M. L. Minsky. *Computation: Finite and Infinite Machines*. Prentice Hall, Inc., 1967.

[MMPV14] 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.

[MMV10] F. Mogavero, A. Murano, and M. Y. Vardi. Reasoning about strategies. In FSTTCS'10, Leibniz International Proceedings in Informatics 8, p. 133–144. Leibniz-Zentrum für Informatik, 2010.

[Pnu77] A. Pnueli. The temporal logic of programs. In FOCS'77, p. 46–57. IEEE Comp. Soc. Press, 1977.

[QS82] J.-P. Queille and J. Sifakis. Specification and verification of concurrent systems in CESAR. In SOP'82, LNCS 137, p. 337–351. Springer, 1982.