

Dependences in Strategy Logic*

Patrick Gardy¹, Patricia Bouyer¹, and Nicolas Markey^{1,2}

1 LSV, CNRS & ENS Paris-Saclay, Univ. Paris-Saclay (France)

2 Univ. Rennes, CNRS, Inria, IRISA (France)

Abstract

Strategy Logic (SL) is a very expressive temporal logic for specifying and verifying properties of multi-agent systems: in SL, one can quantify over strategies, assign them to agents, and express LTL properties of the resulting plays. Such a powerful framework has two drawbacks: first, model checking SL has non-elementary complexity; second, the exact semantics of SL is rather intricate, and may not correspond to what is expected. In this paper, we focus on *strategy dependences* in SL, by tracking how existentially-quantified strategies in a formula may (or may not) depend on other strategies selected in the formula, revisiting the approach of [Mogavero *et al.*, Reasoning about strategies: On the model-checking problem, 2014]. We explain why *elementary* dependences, as defined by Mogavero *et al.*, do not exactly capture the intended concept of behavioral strategies. We address this discrepancy by introducing *timeline* dependences, and exhibit a large fragment of SL for which model checking can be performed in 2-EXPTIME under this new semantics.

1998 ACM Subject Classification F.3.1 Specifying and Verifying and Reasoning about Programs, F.4.1 Mathematical Logic

Keywords and phrases strategic reasoning; strategy logic; dependences; behavioural strategies.

Digital Object Identifier 10.4230/LIPIcs.STACS.2018.34

1 Introduction

Temporal logics. Since Pnueli’s seminal paper [24] in 1977, temporal logics have been widely used in theoretical computer science, especially by the formal-verification community. Temporal logics provide powerful languages for expressing properties of reactive systems, and enjoy efficient algorithms for satisfiability and model checking [9]. Since the early 2000s, new temporal logics have appeared to address *open* and *multi-agent systems*. While classical temporal logics (e.g. CTL [8, 25] and LTL [24]) could only deal with one or all the behaviours of the whole system, ATL [2] expresses properties of (executions generated by) behaviours of individual components of the system. ATL has been extensively studied since then, both about its expressiveness and about its verification algorithms [2, 13, 16].

Strategic interactions in ATL. Strategies in ATL are handled in a very limited way, and there are no real *strategic interactions* in that logic (which, in return, enjoys a polynomial-time model-checking algorithm). Over the last 10 years, various extensions have been defined and studied in order to allow for more interactions [1, 7, 6, 18, 26]. *Strategy Logic* (SL for short) [7, 18] is such a powerful approach, in which strategies are first-class objects; formulas can quantify (universally and existentially) over strategies, store those strategies in variables, assign them to players, and express properties of the resulting plays. As a simple example, the existence of a winning strategy for Player *A* (with objective φ_A) against any strategy of

* Supported by ERC project EQualIS (308087).



Player B would be written as $\exists\sigma_A. \forall\sigma_B. \text{assign}(A \mapsto \sigma_A; B \mapsto \sigma_B). \varphi_A$. This makes the logic both expressive and easy to use (at first sight), at the expense of a very high complexity: SL model checking has non-elementary complexity, and satisfiability is undecidable [18, 15].

Understanding SL. Since it enjoys decidable model checking and high expressiveness, SL is the logic of choice for showing that some game problems are decidable (e.g. rational synthesis [12, 14, 10] or assume-admissible synthesis [5]). For instance, the existence of an admissible strategy for player A (i.e., a strategy that is strictly dominated by no other strategies [5]) is expressed as

$$\exists\sigma_A. \forall\sigma'_A. \left[\begin{array}{l} \exists\sigma_B. (\text{assign}(A \mapsto \sigma_A, B \mapsto \sigma_B). \varphi_A \wedge \text{assign}(A \mapsto \sigma'_A, B \mapsto \sigma_B). \neg\varphi_A) \\ \vee \\ \forall\sigma'_B. (\text{assign}(A \mapsto \sigma_A, B \mapsto \sigma'_B). \varphi_A \vee \text{assign}(A \mapsto \sigma'_A, B \mapsto \sigma'_B). \neg\varphi_A) \end{array} \right]$$

However, it has been noticed in recent works that the nice expressiveness of SL comes with unexpected phenomena. One such phenomenon is induced by the separation of strategy quantification and strategy assignment: are the events between strategy quantifications and strategy assignments part of the *memory* of the strategy? While both options may make sense depending on the applications, only one of them makes model checking decidable [4].

A second phenomenon—which is the main focus of the present paper—concerns *strategy dependences* [18]: in a formula such as $\forall\sigma_A. \exists\sigma_B. \xi$, the existentially-quantified strategy σ_B may depend on *the whole* strategy σ_A ; in other terms, the action returned by strategy σ_B after some finite history ρ may depend on what strategy σ_A would play on any other history ρ' . Again, in some contexts, it may be desirable that the value of strategy σ_B after history ρ can be *computed* based solely on what has been observed along ρ (see Fig. 2 for an illustration). This approach was initiated in [18, 21], conjecturing that large fragments of SL (subsuming ATL *) would have 2-EXPTIME model-checking algorithms with such limited dependences.

Our contributions. We follow this line of work by performing a more thorough exploration of strategy dependences in (a fragment of) SL. We mainly follow the framework of [21], based on a kind of Skolemization of the formula: for instance, a formula of the form $(\forall x_i \exists y_i)_i. \xi$ is satisfied if there exists a *dependence map* θ defining each existentially-quantified strategy y_j based on the universally-quantified strategies $(x_i)_i$. In order to recover the classical semantics of SL, it is only required that the strategy $\theta((x_i)_i)(y_j)$ (i.e. the strategy assigned to the existentially-quantified variable y_j by $\theta((x_i)_i)$) only depends on $(x_i)_{i < j}$.

Based on this definition, other constraints can be imposed on dependence maps, in order to refine the dependences of existentially-quantified strategies on universally-quantified ones. *Elementary dependences* [21] only allows existentially-quantified strategy y_j to depend on the values of $(x_i)_{i < j}$ along the current history. This gives rise to two different semantics in general, but fragments of SL have been defined (SL[1G] in [17], SL[CG] and SL[DG] in [20]) on which the classic and elementary semantics would coincide.

The coincidence actually only holds for SL[1G]. As we explain in this paper, elementary dependences as defined and used in [17, 20] do not exactly capture the intuition that strategies should depend on the “behavior [of universal strategies] on the history of interest only” [20]: indeed, they only allow dependences on universally-quantified strategies *that appear earlier in the formula*, while we claim that the behaviour of all universally-quantified strategies should be considered. We address this discrepancy by introducing another kind of dependences, which we call *timeline dependences*, and which extend elementary dependences by allowing existentially-quantified strategies to additionally depend on *all* universally-quantified strategies along *strict prefixes* of the current history (as illustrated on Fig. 4).

We study and compare those three dependences (classic, elementary and timeline), showing that they correspond to three distinct semantics. Because the semantics based on dependence maps is defined in terms of the *existence* of a witness map, we show that the syntactic negation of a formula may not correspond to its semantic negation: there are cases where both a formula φ and its syntactic negation $\neg\varphi$ fail to hold (i.e., none of them has a witness map). This phenomenon is already present, but had not been formally identified, in [18, 21]. The main contribution of the present paper is the definition of a large (and, in a sense, maximal) fragment of SL for which syntactic and semantic negations coincide under the timeline semantics. As an (important) side result, we show that model checking this fragment under the timeline semantics is \mathcal{L} -EXPTIME-complete.

2 Definitions

2.1 Concurrent game structures

Let AP be a set of atomic propositions, \mathcal{V} be a set of variables, and Agt be a set of agents. A *concurrent game structure* is a tuple $\mathcal{G} = (\text{Act}, Q, \Delta, \text{lab})$ where Act is a finite set of actions, Q is a finite set of states, $\Delta: Q \times \text{Act}^{\text{Agt}} \rightarrow Q$ is the transition function, and $\text{lab}: Q \rightarrow 2^{\text{AP}}$ is a labelling function. An element of Act^{Agt} will be called a *move vector*. For any $q \in Q$, we let $\text{succ}(q)$ be the set $\{q' \in Q \mid \exists m \in \text{Act}^{\text{Agt}}. q' = \Delta(q, m)\}$. For the sake of simplicity, we assume in the sequel that $\text{succ}(q) \neq \emptyset$ for any $q \in Q$. A game \mathcal{G} is said *turn-based* whenever for every state $q \in Q$, there is a player $\text{own}(\cdot)q \in \text{Agt}$ (named the *owner* of q) such that for any two move vectors m_1 and m_2 with $m_1(\text{own}(\cdot)q) = m_2(\text{own}(\cdot)q)$, it holds $\Delta(q, m_1) = \Delta(q, m_2)$. Figure 1 displays an example of a (turn-based) game.

Fix a state $q \in Q$. A *play* in \mathcal{G} from q is an infinite sequence $\pi = (q_i)_{i \in \mathbb{N}}$ of states in Q such that $q_0 = q$ and $q_i \in \text{succ}(q_{i-1})$ for all $i > 0$. We write $\text{Play}_{\mathcal{G}}(q)$ for the set of plays in \mathcal{G} from q . In this and all similar notations, we might omit to mention \mathcal{G} when it is clear from the context, and q when we consider the union over all $q \in Q$. A (strict) prefix of a play π is a finite sequence $\rho = (q_i)_{0 \leq i \leq L}$, for some $L \in \mathbb{N}$. We write $\text{Pref}(\pi)$ for the set of strict prefixes of play π . Such finite prefixes are called *histories*, and we let $\text{Hist}_{\mathcal{G}}(q) = \text{Pref}(\text{Play}_{\mathcal{G}}(q))$. We extend the notion of strict prefixes and the notation Pref to histories in the natural way, requiring in particular that $\rho \notin \text{Pref}(\rho)$. A (finite) extension of a history ρ is any history ρ' such that $\rho \in \text{Pref}(\rho')$. Let $\rho = (q_i)_{i \leq L}$ be a history. We define $\text{first}(\rho) = q_0$ and $\text{last}(\rho) = q_L$. Let $\rho' = (q'_j)_{j \leq L'}$ be a history from $\text{last}(\rho)$. The *concatenation* of ρ and ρ' is then defined as the path $\rho \cdot \rho' = (q''_k)_{k \leq L+L'}$ such that $q''_k = q_k$ when $k \leq L$ and $q''_k = q'_{k-L}$ when $L \geq k$ (notice that we required $q'_0 = q_L$).

A *strategy* from q is a mapping $\delta: \text{Hist}_{\mathcal{G}}(q) \rightarrow \text{Act}$. We write $\text{Strat}_{\mathcal{G}}(q)$ for the set of strategies in \mathcal{G} from q . Given a strategy $\delta \in \text{Strat}(q)$ and a history ρ from q , the *translation* $\delta_{\vec{\rho}}$ of δ by ρ is the strategy $\delta_{\vec{\rho}}$ from $\text{last}(\rho)$ defined by $\delta_{\vec{\rho}}(\rho') = \delta(\rho \cdot \rho')$ for any $\rho' \in \text{Hist}(\text{last}(\rho))$. A *valuation* from q is a partial function $\chi: \mathcal{V} \cup \text{Agt} \rightarrow \text{Strat}(q)$. As usual, for any partial function f , we write $\text{dom}(f)$ for the domain of f .

Let $q \in Q$ and χ be a valuation from q . If $\text{Agt} \subseteq \text{dom}(\chi)$, then χ induces a unique play from q , called its *outcome*, and defined as $\text{out}(q, \chi) = (q_i)_{i \in \mathbb{N}}$ such that $q_0 = q$ and for every $i \in \mathbb{N}$, we have $q_{i+1} = \Delta(q_i, m_i)$ with $m_i(A) = \chi(A)((q_j)_{j \leq i})$ for every $A \in \text{Agt}$.

2.2 Strategy Logic with boolean goals

Strategy Logic (SL for short) was introduced in [7], and further extended and studied in [22, 18], as a rich logical formalism for expressing properties of games. SL manipulates

strategies as first-order elements, assigns them to players, and expresses LTL properties on the outcomes of the resulting strategic interactions. This results in a very expressive temporal logic, for which satisfiability is undecidable [22, 19] and model checking is TOWER-complete [18, 3]. In this paper, we focus on a restricted fragment of SL, called $\text{SL}[\text{BG}]^b$ (where BG stands for *boolean goals* [18], and the symbol b indicates that we do not allow nesting of (closed) subformulas; we discuss this latter restriction below).

Syntax. Formulas in $\text{SL}[\text{BG}]^b$ are built along the following grammar

$$\begin{aligned} \text{SL}[\text{BG}]^b \ni \varphi ::= \exists x. \varphi \mid \forall x. \varphi \mid \xi & \quad \xi ::= \neg \xi \mid \xi \wedge \xi \mid \xi \vee \xi \mid \beta \\ \beta ::= \text{assign}(\sigma). \psi & \quad \psi ::= \neg \psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathbf{X} \psi \mid \psi \mathbf{U} \psi \mid p \end{aligned}$$

where x ranges over \mathcal{V} , σ ranges over the set \mathcal{V}^{Agt} of *full assignments*, and p ranges over AP . A *goal* is a formula of the form β in the grammar above; it expresses an LTL property ψ on the outcome of the mapping σ . Formulas in $\text{SL}[\text{BG}]^b$ are thus made of an initial block of first-order quantifiers (selecting strategies for variables in \mathcal{V}), followed by a boolean combination of goals.

Free variables. With any subformula ζ of some formula $\varphi \in \text{SL}[\text{BG}]^b$, we associate its set of *free agents and variables*, which we write $\text{free}(\zeta)$. It contains the agents and variables that have to be associated with a strategy in order to unequivocally evaluate ζ (as will be seen from the definition of the semantics of $\text{SL}[\text{BG}]^b$ below). The set $\text{free}(\zeta)$ is defined inductively:

$$\begin{aligned} \text{free}(p) &= \emptyset \quad \text{for all } p \in \text{AP} & \text{free}(\mathbf{X} \psi) &= \text{Agt} \cup \text{free}(\psi) \\ \text{free}(\neg \alpha) &= \text{free}(\alpha) & \text{free}(\psi_1 \mathbf{U} \psi_2) &= \text{Agt} \cup \text{free}(\psi_1) \cup \text{free}(\psi_2) \\ \text{free}(\alpha_1 \vee \alpha_2) &= \text{free}(\alpha_1) \cup \text{free}(\alpha_2) & \text{free}(\exists x. \varphi) &= \text{free}(\varphi) \setminus \{x\} \\ \text{free}(\alpha_1 \wedge \alpha_2) &= \text{free}(\alpha_1) \cup \text{free}(\alpha_2) & \text{free}(\forall x. \varphi) &= \text{free}(\varphi) \setminus \{x\} \\ \text{free}(\text{assign}(\sigma). \varphi) &= (\text{free}(\varphi) \cup \sigma(\text{Agt} \cap \text{free}(\varphi))) \setminus \text{Agt} \end{aligned}$$

Subformula ζ is said to be *closed* whenever $\text{free}(\zeta) = \emptyset$. We can now comment on our choice of considering the flat fragment of $\text{SL}[\text{BG}]$: the full fragment, as defined in [18], allows for nesting *closed* $\text{SL}[\text{BG}]$ formulas in place of atomic propositions. The meaning of such nesting in our setting is ambiguous, because our semantics (in Sections 3 to 5) are defined in terms of the *existence of a witness*, which does not easily propagate in formulas. In particular, as we explain later in the paper, the semantics of the negation of a formula (there is a witness for $\neg \varphi$) does not coincide with the negation of the semantics (there is no witness for φ); thus substituting a subformula and substituting its negation may return different results.

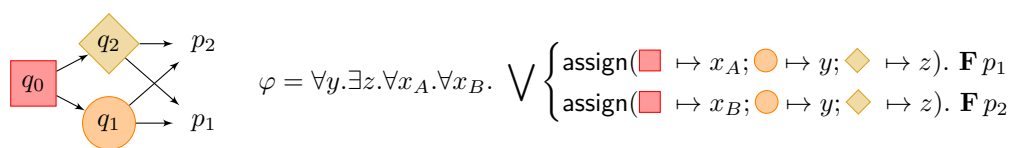
Semantics. Fix a state $q \in Q$, and a valuation $\chi: \mathcal{V} \cup \text{Agt} \rightarrow \text{Strat}(q)$. We inductively define the semantics of a subformula α of a formula of $\text{SL}[\text{BG}]^b$ at q under valuation χ , requiring $\text{free}(\alpha) \subseteq \text{dom}(\chi)$. We omit the easy cases of boolean combinations and atomic propositions.

Given a mapping $\sigma: \text{Agt} \rightarrow \mathcal{V}$, the semantics of strategy assignments is defined as follows:

$$\mathcal{G}, q \models_{\chi} \text{assign}(\sigma). \psi \quad \Leftrightarrow \quad \mathcal{G}, q \models_{\chi[A \in \text{Agt} \mapsto \chi(\sigma(A))]} \psi.$$

Notice that, writing $\chi' = \chi[A \in \text{Agt} \mapsto \chi(\sigma(A))]$, we have $\text{free}(\psi) \subseteq \text{dom}(\chi')$ if $\text{free}(\alpha) \subseteq \text{dom}(\chi)$, so that our inductive definition is sound.

We now consider path formulas $\psi = \mathbf{X} \psi_1$ and $\psi = \psi_1 \mathbf{U} \psi_2$. Since $\text{Agt} \subseteq \text{free}(\psi) \subseteq \text{dom}(\chi)$, the valuation χ induces a unique outcome $\text{out}(q, \chi) = (q_i)_{i \in \mathbb{N}}$ from q . For $n \in \mathbb{N}$, we write $\text{out}_n(q, \chi) = (q_i)_{i \leq n}$, and define $\chi_{\vec{n}}$ as the valuation obtained by shifting all



■ **Figure 1** A game and a SL[BG] formula.

the strategies in the image of χ by $\text{out}_n(q, \chi)$. Under the same conditions, we also define $q_{\vec{n}} = \text{last}(\text{out}_n(q, \chi))$. We then set

$$\begin{aligned} \mathcal{G}, q \models_{\chi} \mathbf{X} \psi_1 &\Leftrightarrow \mathcal{G}, q_{\vec{1}} \models_{\chi_{\vec{1}}} \psi_1 \\ \mathcal{G}, q \models_{\chi} \psi_1 \mathbf{U} \psi_2 &\Leftrightarrow \exists k \in \mathbb{N}. \mathcal{G}, q_{\vec{k}} \models_{\chi_{\vec{k}}} \psi_2 \quad \text{and} \quad \forall 0 \leq j < k. \mathcal{G}, q_{\vec{j}} \models_{\chi_{\vec{j}}} \psi_1. \end{aligned}$$

In the sequel, we use classical shorthands, such as \top for $p \vee \neg p$ (for any $p \in \text{AP}$), $\mathbf{F} \psi$ for $\top \mathbf{U} \psi$ (*eventually* ψ), and $\mathbf{G} \psi$ for $\neg \mathbf{F} \neg \psi$ (*always* ψ). It remains to define the semantics of the strategy quantifiers. This is actually what this paper is all about. We provide here the original semantics, and discuss alternatives in the following sections:

$$\mathcal{G}, q \models_{\chi} \exists x. \varphi \Leftrightarrow \exists \delta \in \text{Strat}(q). \mathcal{G}, q \models_{\chi[x \mapsto \delta]} \varphi.$$

► **Example 1.** We consider the (turn-based) game \mathcal{G} is depicted on Fig. 1. We name the players after the shape of the state they control. The SL[BG] formula φ to the right of Fig. 1 has four quantified variables and two goals. We show that this formula evaluates to true at q_0 : fix a strategy δ_y (to be played by player \circ); because \mathcal{G} is turn-based, we identify the actions of the owner of a state with the resulting target state, so that $\delta_y(q_0 q_1)$ will be either p_1 or p_2 . We then define strategy δ_z (to be played by \diamond) as $\delta_z(q_0 q_2) = \delta_y(q_0 q_1)$. Then clearly, for any strategy assigned to player \square , one of the goals of formula φ holds true, so that φ itself evaluates to true.

Subclasses of SL[BG]. Because of the high complexity and subtlety of reasoning with SL and SL[BG], several restrictions of SL[BG] have been considered in the literature [17, 20, 21], by adding further restrictions to boolean combinations in the grammar defining the syntax:

- SL[1G] restricts SL[BG] to a unique goal. $\text{SL}[1G]^b$ is then defined from the grammar of $\text{SL}[BG]^b$ by setting $\xi ::= \beta$ in the grammar;
- the larger fragment SL[CG] allows for *conjunctions* of goals. $\text{SL}[CG]^b$ corresponds to formulas defined with $\xi ::= \xi \wedge \xi \mid \beta$;
- similarly, SL[DG] only allows *disjunctions* of goals, i.e. $\xi ::= \xi \vee \xi \mid \beta$;
- finally, SL[AG] mixes conjunctions and disjunctions in a restricted way. Goals in $\text{SL}[AG]^b$ can be combined using the following grammar: $\xi ::= \beta \wedge \xi \mid \beta \vee \xi \mid \beta$.

In the sequel, we write a generic $\text{SL}[BG]^b$ formula φ as $(Q_i x_i)_{1 \leq i \leq l}. \xi(\beta_j. \psi_j)_{j \leq n}$ where:

- $(Q_i x_i)_{1 \leq i \leq l}$ is a block of quantifications, with $\{x_i \mid 1 \leq i \leq l\} \subseteq \mathcal{V}$ and $Q_i \in \{\exists, \forall\}$, for every $1 \leq i \leq l$;
- $\xi(g_1, \dots, g_n)$ is a boolean combination of its arguments;
- for all $1 \leq j \leq n$, $\beta_j. \psi_j$ is a goal: β_j is a full assignment and ψ_j is an LTL formula.

3 Strategy dependences

We now follow the framework of [18, 21] and define the semantics of $\text{SL}[BG]^b$ in terms of *dependence maps*. This approach provides a fine way of controlling how *existentially-quantified*

strategies depend on other strategies (in a quantifier block). Using dependence maps, we can limit such dependences.

Dependence maps. Consider an $\text{SL}[\text{BG}]^b$ formula $\varphi = (Q_i x_i)_{1 \leq i \leq l} \cdot \xi(\beta_j \cdot \varphi_j)_{j \leq n}$, assuming w.l.o.g. that $\{x_i \mid 1 \leq i \leq l\} = \mathcal{V}$. We let $\mathcal{V}^\forall = \{x_i \mid Q_i = \forall\} \subseteq \mathcal{V}$ be the set of universally-quantified variables of φ . A function $\theta: \text{Strat}^{\mathcal{V}^\forall} \rightarrow \text{Strat}^{\mathcal{V}}$ is a φ -map (or map when φ is clear from the context) if $\theta(w)(x_i)(\rho) = w(x_i)(\rho)$ for any $w \in \text{Strat}^{\mathcal{V}^\forall}$, any $x_i \in \mathcal{V}^\forall$, and any history ρ . In other words, $\theta(w)$ extends w to \mathcal{V} . This general notion allows any existentially-quantified variable to depend on *all* universally-quantified ones (dependence on existentially-quantified variables is implicit: all existentially-quantified variables are assigned through a single map, hence they all depend on the others); we add further restrictions later on. Using maps, we may then define new semantics for $\text{SL}[\text{BG}]^b$: generally speaking, formula $\varphi = (Q_i x_i)_{1 \leq i \leq l} \cdot \xi(\beta_j \cdot \varphi_j)_{j \leq n}$ holds true if there exists a φ -map θ such that, for any $w: \mathcal{V}^\forall \rightarrow \text{Strat}$, the valuation $\theta(w)$ makes $\xi(\beta_j \cdot \varphi_j)_{j \leq n}$ hold true.

Classic maps are dependence maps in which the order of quantification is respected:

$$\forall w_1, w_2 \in \text{Strat}^{\mathcal{V}^\forall}. \forall x_i \in \mathcal{V} \setminus \mathcal{V}^\forall. \\ (\forall x_j \in \mathcal{V}^\forall \cap \{x_j \mid j < i\}. w_1(x_j) = w_2(x_j)) \Rightarrow (\theta(w_1)(x_i) = \theta(w_2)(x_i)). \quad (\text{C})$$

In words, if w_1 and w_2 coincide on $\mathcal{V}^\forall \cap \{x_j \mid j < i\}$, then $\theta(w_1)$ and $\theta(w_2)$ coincide on x_i .

Elementary maps [18, 17] have to satisfy a more restrictive condition: for those maps, the value of an existentially-quantified strategy at any history ρ may only depend on the value of earlier universally-quantified strategies *along* ρ . This may be written as:

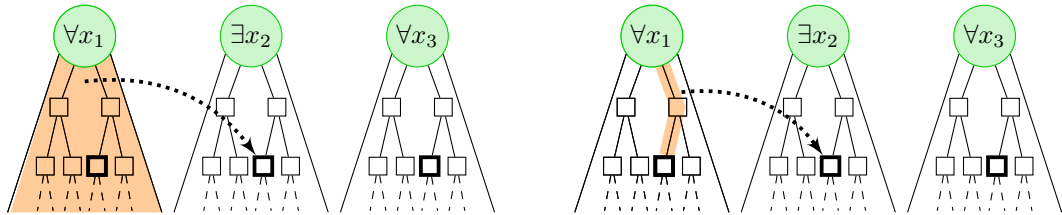
$$\forall w_1, w_2 \in \text{Strat}^{\mathcal{V}^\forall}. \forall x_i \in \mathcal{V} \setminus \mathcal{V}^\forall. \forall \rho \in \text{Hist}. \\ (\forall x_j \in \mathcal{V}^\forall \cap \{x_k \mid k < i\}. \forall \rho' \in \text{Pref}(\rho) \cup \{\rho\}. w_1(x_j)(\rho') = w_2(x_j)(\rho')) \Rightarrow \\ (\theta(w_1)(x_i)(\rho) = \theta(w_2)(x_i)(\rho)). \quad (\text{E})$$

In this case, for any history ρ , if two valuations w_1 and w_2 of the universally-quantified variables coincide on the variables quantified before x_i all along ρ , then $\theta(w_1)(x_i)$ and $\theta(w_2)(x_i)$ have to coincide at ρ .

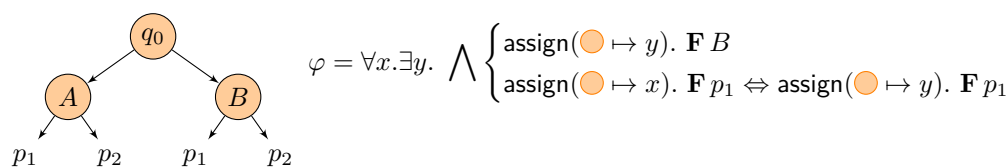
The difference between both kinds of dependences is illustrated on Fig. 2: for classic maps, the existentially-quantified strategy x_2 may depend on the whole strategy x_1 , while it may only depend on the value of x_1 along the current history for elementary maps. Notice that a map satisfying (E) also satisfies (C).

Satisfaction relations. Pick a formula $\varphi = (Q_i x_i)_{1 \leq i \leq l} \cdot \xi(\beta_j \cdot \varphi_j)_{j \leq n}$ in $\text{SL}[\text{BG}]^b$. We define:

$$\mathcal{G}, q \models^C \varphi \quad \text{iff} \quad \exists \theta \text{ satisfying (C)}. \forall w \in \text{Strat}^{\mathcal{V}^\forall}. \mathcal{G}, q \models_{\theta(w)} \xi(\beta_j \varphi_j)_{j \leq n}$$



■ **Figure 2** Classical (left) vs elementary (right) dependences for a formula $\forall x_1. \exists x_2. \forall x_3. \xi$



■ **Figure 3** A game \mathcal{G} and an $\text{SL}[\text{BG}]^b$ formula φ such that $\mathcal{G}, q_0 \not\models^E \varphi$ and $\mathcal{G}, q_0 \not\models^E \neg\varphi$.

As explained above, this actually corresponds to the usual semantics of $\text{SL}[\text{BG}]^b$ as given in Section 2 [18, Theorem 4.6]. When $\mathcal{G}, q \models^C \varphi$, a map θ satisfying the conditions above is called a *C-witness* of φ for \mathcal{G} and q . Similarly, we define the *elementary semantics* [18] as:

$$\mathcal{G}, q \models^E \varphi \quad \text{iff} \quad \exists \theta \text{ satisfying (E)}. \forall w \in \text{Strat}^{\forall}. \mathcal{G}, q \models_{\theta(w)} \xi(\beta_j \varphi_j)_{j \leq n}$$

Again, when such a map exists, it is called an *E-witness*. Notice that since Property (E) implies Property (C), we have $\mathcal{G}, q \models^E \varphi \Rightarrow \mathcal{G}, q \models^C \varphi$ for any $\varphi \in \text{SL}[\text{BG}]^b$. This corresponds to the intuition that it is harder to satisfy a $\text{SL}[\text{BG}]^b$ formula when dependences are more restricted. The contrapositive statement then raises questions about the negation of formulas.

The syntactic vs. semantic negations. If $\varphi = (Q_i x_i)_{1 \leq i \leq l} \xi(\beta_j \varphi_j)_{j \leq n}$ is an $\text{SL}[\text{BG}]^b$ formula, its syntactic negation $\neg\varphi$ is the formula $(\overline{Q}_i x_i)_{i \leq l} (\neg\xi)(\beta_j \varphi_j)_{j \leq n}$, where $\overline{Q}_i = \exists$ if $Q_i = \forall$ and $\overline{Q}_i = \forall$ if $Q_i = \exists$. Looking at the definitions of \models^C and \models^E , it could be the case that e.g. $\mathcal{G}, q \models^C \varphi$ and $\mathcal{G}, q \models^C \neg\varphi$: this only requires the existence of two adequate maps. However, since \models^C and \models coincide, and since $\mathcal{G}, q \models \varphi \Leftrightarrow \mathcal{G}, q \not\models \neg\varphi$ in the usual semantics, we get $\mathcal{G}, q \models^C \varphi \Leftrightarrow \mathcal{G}, q \not\models^C \neg\varphi$. Also, since $\mathcal{G}, q \models^E \varphi \Rightarrow \mathcal{G}, q \models^C \varphi$, we also get $\mathcal{G}, q \models^E \varphi \Rightarrow \mathcal{G}, q \not\models^E \neg\varphi$. As we now show, the converse implication holds for $\text{SL}[\text{1G}]^b$, but may fail to hold for $\text{SL}[\text{BG}]^b$.

► **Proposition 1.** *There exist a (one-player) game \mathcal{G} with initial state q_0 and a formula $\varphi \in \text{SL}[\text{BG}]^b$ such that $\mathcal{G}, q_0 \not\models^E \varphi$ and $\mathcal{G}, q_0 \not\models^E \neg\varphi$.*

Proof. Consider the formula and the one-player game of Fig. 3. We start by proving that $\mathcal{G}, q_0 \not\models^E \varphi$. For a contradiction, assume that a witness map θ satisfying (E) exists, and pick any valuation w for the universal variable x . First, for the first goal in the conjunction to be fulfilled, the strategy assigned to y must play to B from q_0 . We abbreviate this as $\theta(w)(y)(q_0) = B$ in the sequel. Now, consider two valuations w_1 and w_2 such that $w_1(x)(q_0) = w_2(x)(q_0) = A$ and $w_1(x)(q_0 \cdot B) = w_2(x)(q_0 \cdot B)$, but such that $w_1(x)(q_0 \cdot A) = p_1$ and $w_2(x)(q_0 \cdot A) = p_2$. In order to fulfill the second goal under both valuations w_1 and w_2 , we must have $\theta(w_1)(y)(q_0 \cdot B) = p_1$ and $\theta(w_2)(y)(q_0 \cdot B) = p_2$. But this violates Property (E): since $w_1(x)$ and $w_2(x)$ coincide on q_0 and on $q_0 \cdot B$, we must have $\theta(w_1)(y)(q_0 \cdot B) = \theta(w_2)(y)(q_0 \cdot B)$.

We now prove that $\mathcal{G}, q_0 \not\models^E \neg\varphi$. Indeed, following the previous discussion, we easily get that $\mathcal{G}, q_0 \models^C \varphi$, by letting $\theta(w)(y)(q_0) = B$ and $\theta(w)(y)(q_0 \cdot B) = w(x)(q_0 \cdot A)$ if $w(x)(q_0) = A$, and $\theta(w)(y)(q_0 \cdot B) = w(x)(q_0 \cdot B)$ if $w(x)(q_0) = B$. As explained above, this entails $\mathcal{G}, q_0 \not\models^C \neg\varphi$, and $\mathcal{G}, q_0 \not\models^E \neg\varphi$. ◀

► **Proposition 2.** *For any game \mathcal{G} with initial state q_0 , and any formula $\varphi \in \text{SL}[\text{1G}]^b$, it holds $\mathcal{G}, q_0 \models^E \varphi \Leftrightarrow \mathcal{G}, q_0 \not\models^E \neg\varphi$.*

Sketch of proof. This result follows from [18, Corollary 4.21], which states that \models^C and \models^E coincide on $\text{SL}[\text{1G}]$. Because it is central in our approach, we sketch a direct proof here

using similar ingredients: it consists in encoding the problem whether $\mathcal{G}, q_0 \models^E \varphi$ into a two-player turn-based game with a parity-winning objective.

The construction is as follows: the interaction between existential and universal quantifications of the formula is integrated into the game structure, replacing each state of \mathcal{G} with a tree-shaped subgame where Player P_{\exists} selects existentially-quantified actions and Player P_{\forall} selects universally-quantified ones. The unique goal of the formula is then incorporated into the game via a deterministic parity automaton, yielding a two-player turn-based parity game. We then show that $\mathcal{G}, q_0 \models^E \varphi$ if, and only if, Player P_{\exists} has a winning strategy in the resulting turn-based parity game, while $\mathcal{G}, q_0 \models^E \neg\varphi$ if, and only if, Player P_{\forall} has a winning strategy. Those equivalences hold for the elementary semantics because memoryless strategies are sufficient in parity games. Proposition 2 then follows by determinacy of those games [11, 23]. ◀

Note that the construction of the parity game gives an effective algorithm for the model-checking problem of $\text{SL}[1\text{G}]^b$, which runs in time doubly-exponential in the size of the formula, and polynomial in the size of the game structure; we recover the result of [18] for that problem.

Comparison of \models^C and \models^E . A consequence of Prop. 2 is that \models^C and \models^E coincide on $\text{SL}[1\text{G}]^b$ (Corollary 4.21 of [18]). However, when considering larger fragments, the satisfaction relations are distinct (see the proof of Prop. 1 for a candidate formula in $\text{SL}[\text{CG}]^b$):

► **Proposition 3.** *The relations \models^C and \models^E differ on $\text{SL}[\text{CG}]^b$, as well as on $\text{SL}[\text{DG}]^b$.*

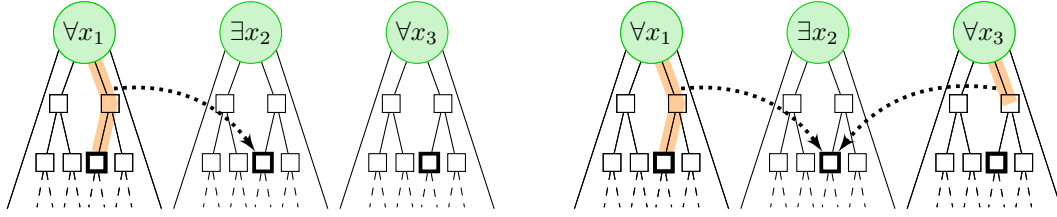
► **Remark.** Proposition 3 contradicts the claim in [20] that \models^E and \models^C coincide on $\text{SL}[\text{CG}]$ (and $\text{SL}[\text{DG}]$). Indeed, in [20], the satisfaction relation for $\text{SL}[\text{DG}]$ and $\text{SL}[\text{CG}]$ is encoded into a two-player game in pretty much the same way as we did in the proof of Prop. 2. While this indeed rules out dependences outside the current history, it also gives information to Player P_{\exists} about the values (over prefixes of the current history) of strategies that are universally-quantified later in the quantification block. This proof technique works with $\text{SL}[1\text{G}]^b$ because the single goal can be encoded as a parity objective, for which memoryless strategies exist, so that the extra information is not crucial. In the next section, we investigate the role of this extra information for larger fragments of $\text{SL}[\text{BG}]^b$.

4 Timeline dependences

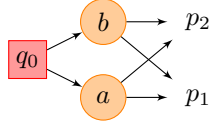
Following the discussion above, we introduce a new type of dependences between strategies (which we call *timeline dependences*). They allow strategies to also observe (and depend on) *all* other universally-quantified strategies on the strict prefix of the current history. For instance, for a block of quantifiers $\forall x_1. \exists x_2. \forall x_3$, the value of x_2 after history ρ may depend on the value of x_1 on ρ and its prefixes (as for elementary maps), but also on the value of x_3 on the (strict) prefixes of ρ . Such dependences are depicted on Fig. 4. We believe that such dependences are relevant in many situations, especially for reactive synthesis, since in this framework strategies really base their decisions on what they could observe along the current history.

Formally, a map θ is a *timeline map* if it satisfies the following condition:

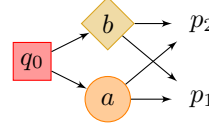
$$\forall w_1, w_2 \in \text{Strat}^{\mathcal{V}^{\forall}}. \forall x_i \in \mathcal{V} \setminus \mathcal{V}^{\forall}. \forall \rho \in \text{Hist}. \\ \left(\begin{array}{l} \forall x_j \in \mathcal{V}^{\forall} \cap \{x_k \mid k < i\}. \forall \rho' \in \text{Pref}(\rho) \cup \{\rho\}. w_1(x_j)(\rho) = w_2(x_j)(\rho) \\ \wedge \forall x_j \in \mathcal{V}^{\forall}. \forall \rho' \in \text{Pref}(\rho). w_1(x_j)(\rho) = w_2(x_j)(\rho) \end{array} \right) \Rightarrow \\ (\theta(w_1)(x_i)(\rho) = \theta(w_2)(x_i)(\rho)). \quad (\text{T})$$



■ **Figure 4** Elementary (left) vs timeline (right) dependences for a formula $\forall x_1. \exists x_2. \forall x_3. \xi$



■ **Figure 5** \models^E and \models^T differ on $SL[CG]^b$



■ **Figure 6** \models^E and \models^T differ on $SL[DG]^b$

Using those maps, we introduce the *timeline semantics* of $SL[BG]^b$:

$$\mathcal{G}, q \models^T \varphi \text{ iff } \exists \theta \text{ satisfying (T)}. \forall w \in \text{Strat}^{\forall}. \mathcal{G}, q \models_{\theta(w)} \xi(\beta_j \varphi_j)_{j \leq n}$$

Such a map, if any, is called a *T-witness* of φ for \mathcal{G} and q . As in the previous section, it is easily seen that Property (E) implies Property (T), so that an E-witness is also a T-witness, and $\mathcal{G}, q \models^E \varphi \Rightarrow \mathcal{G}, q \models^T \varphi$ for any formula $\varphi \in SL[BG]^b$.

► **Example 2.** Consider again the game of Fig 1 in Section 2. We have seen that $\mathcal{G}, q_0 \models^C \varphi$ in Section 2, and that $\mathcal{G}, q_0 \not\models^E \varphi$ in the proof of Prop. 3. With timeline dependences, we have $\mathcal{G}, q_0 \models^T \varphi$. Indeed, now $\theta(w)(z)(q_0 \cdot q_2)$ may depend on $w(x_A)(q_0)$ and $w(x_B)(q_0)$: we could then have e.g. $\theta(w)(z)(q_0 \cdot q_2) = p_1$ when $w(x_A)(q_0) = q_2$, and $\theta(w)(z)(q_0 \cdot q_2) = p_2$ when $w(x_A)(q_0) = q_1$. It is easily checked that this map is a T-witness of φ for q_0 .

Comparison of \models^E and \models^T . As explained at the end of Section 3, the proof of Prop. 2 actually shows the following result:

► **Proposition 4.** For any game \mathcal{G} with initial state q_0 , and any formula $\varphi \in SL[IG]^b$, it holds $\mathcal{G}, q_0 \models^E \varphi \Leftrightarrow \mathcal{G}, q_0 \models^T \varphi$.

We now prove that this does not extend to $SL[CG]^b$ and $SL[DG]^b$:

► **Proposition 5.** The relations \models^E and \models^T differ on $SL[CG]^b$, as well as on $SL[DG]^b$.

Proof. For $SL[CG]^b$, we consider the game structure of Fig. 5, and formula

$$\varphi_C = \exists y. \forall x_A. \exists x_B. \bigwedge \begin{cases} \text{assign}(\text{orange} \mapsto y; \text{red} \mapsto x_A). \mathbf{F} p_1 \\ \text{assign}(\text{orange} \mapsto y; \text{red} \mapsto x_B). \mathbf{F} p_2 \end{cases}$$

We first notice that $\mathcal{G}, q_0 \not\models^E \varphi_C$: indeed, in order to satisfy the first goal under any choice of x_A , the strategy for y has to point to p_1 from both a and b . But then no choice of x_B will make the second goal true.

On the other hand, considering the timeline semantics, strategy y after $q_0 \cdot a$ and $q_0 \cdot b$ may depend on the choice of x_A in q_0 . When $w(x_A)(q_0) = a$, we let $\theta(w)(y)(q_0 \cdot a) = p_1$ and $\theta(w)(y)(q_0 \cdot b) = p_2$ and $\theta(w)(x_B)(q_0) = b$, which makes both goals hold true. Conversely, if $w(x_A)(q_0) = b$, then we let $\theta(w)(y)(q_0 \cdot b) = p_1$ and $\theta(w)(y)(q_0 \cdot a) = p_2$ and $\theta(w)(x_B)(q_0) = a$.

For $\text{SL}[\text{DG}]^b$, we consider the game of Fig. 6, and easily prove that formula φ_D below has a T-witness but no E-witness:

$$\varphi_D = \exists y. \forall x_A. \forall x_B. \forall z. \bigvee \left\{ \begin{array}{l} \text{assign}(\text{orange} \mapsto y; \text{red} \mapsto x_A; \text{yellow} \mapsto z). \mathbf{F} p_1 \\ \text{assign}(\text{orange} \mapsto y; \text{red} \mapsto x_B; \text{yellow} \mapsto z). \mathbf{F} p_2 \end{array} \right. \blacktriangleleft$$

The syntactic vs. semantic negations. While both semantics differ, we now prove that the situation w.r.t. the syntactic vs. semantic negations is similar. First, following Prop. 4 and 2, the two negations coincide on $\text{SL}[\text{1G}]^b$ under the timeline semantics. Moreover:

► **Proposition 6.** *For any formula φ in $\text{SL}[\text{BG}]^b$, for any game \mathcal{G} and any state q_0 , we have $\mathcal{G}, q_0 \models^T \varphi \Rightarrow \mathcal{G}, q_0 \not\models^T \neg\varphi$.*

Sketch of proof. Write $\varphi = (Q_i x_i)_{1 \leq i \leq l} \xi(\beta_j \varphi_j)_{j \leq n}$. For a contradiction, assume that there exist two maps θ and $\bar{\theta}$ witnessing $\mathcal{G}, q_0 \models^T \varphi$ and $\mathcal{G}, q_0 \models^T \neg\varphi$, respectively. Then for any strategy valuations w and \bar{w} for \mathcal{V}^\forall and \mathcal{V}^\exists , we have that $\mathcal{G}, q_0 \models_{\theta(w)} \xi(\beta_j \varphi_j)_j$ and $\mathcal{G}, q_0 \models_{\bar{\theta}(\bar{w})} \neg\xi(\beta_j \varphi_j)_j$. We can then inductively (on histories and on the sequence of quantified variables) build a strategy valuation χ on \mathcal{V} such that $\theta(\chi|_{\mathcal{V}^\forall}) = \bar{\theta}(\chi|_{\mathcal{V}^\exists}) = \chi$. Then under valuation χ , both $\xi(\beta_j \varphi_j)_j$ and $\neg\xi(\beta_j \varphi_j)_j$ hold in q_0 , which is impossible. ◀

► **Proposition 7.** *There exists a formula $\varphi \in \text{SL}[\text{BG}]^b$, a (turn-based) game \mathcal{G} and a state q_0 such that $\mathcal{G}, q_0 \not\models^T \varphi$ and $\mathcal{G}, q_0 \not\models^T \neg\varphi$.*

5 The fragment $\text{SL}[\text{EG}]^b$

In this section, we focus on the timeline semantics \models^T . We exhibit a fragment¹ $\text{SL}[\text{EG}]^b$ of $\text{SL}[\text{BG}]^b$, containing $\text{SL}[\text{CG}]^b$ and $\text{SL}[\text{DG}]^b$, for which the syntactic and semantic negations coincide, and for which we prove model-checking is in 2-EXPTIME:

► **Theorem 8.** *For any $\varphi \in \text{SL}[\text{EG}]^b$ and any state q_0 , it holds: $\mathcal{G}, q_0 \models^T \varphi \Leftrightarrow \mathcal{G}, q_0 \not\models^T \neg\varphi$. Moreover, model checking $\text{SL}[\text{EG}]^b$ for the timeline semantics is 2-EXPTIME-complete.*

5.1 Semi-stable sets.

For $n \in \mathbb{N}$, we let $\{0, 1\}^n$ be the set of mappings from $[1, n]$ to $\{0, 1\}$. We write $\mathbf{0}^n$ (or $\mathbf{0}$ if the size n is clear) for the function that maps all integers in $[1, n]$ to 0, and $\mathbf{1}^n$ (or $\mathbf{1}$) for the function that maps $[1, n]$ to 1. For $f, g \in \{0, 1\}^n$, we define:

$$\bar{f}: i \mapsto 1 - f(i) \quad f \wedge g: i \mapsto \min\{f(i), g(i)\} \quad f \vee g: i \mapsto \max\{f(i), g(i)\}.$$

We then introduce the notion of semi-stable sets, on which the definition of $\text{SL}[\text{EG}]^b$ relies: a set $F^n \subseteq \{0, 1\}^n$ is *semi-stable* if for any f and g in F^n , it holds that

$$\forall s \in \{0, 1\}^n. \quad (f \wedge s) \vee (g \wedge \bar{s}) \in F^n \text{ or } (g \wedge s) \vee (f \wedge \bar{s}) \in F^n.$$

► **Example 3.** *Obviously, the set $\{0, 1\}^n$ is semi-stable, as well as the empty set. For $n = 2$, the set $\{(0, 1), (1, 0)\}$ is easily seen not to be semi-stable: taking $f = (0, 1)$ and $g = (1, 0)$ with $s = (1, 0)$, we get $(f \wedge s) \vee (g \wedge \bar{s}) = (0, 0)$ and $(g \wedge s) \vee (f \wedge \bar{s}) = (1, 1)$. Similarly, $\{(0, 0), (1, 1)\}$ is not semi-stable. Any other subset of $\{0, 1\}^2$ is semi-stable.*

¹ We name our fragment $\text{SL}[\text{EG}]^b$ as it comes as a natural continuation after fragments $\text{SL}[\text{AG}]^b$ [21], $\text{SL}[\text{BG}]^b$ [18], and $\text{SL}[\text{CG}]^b$ and $\text{SL}[\text{DG}]^b$ [20].

We then define

$$\begin{aligned} \text{SL}[\text{EG}]^b \ni \varphi &::= \forall x. \varphi \mid \exists x. \varphi \mid \xi & \xi &::= F^n((\beta_i)_{1 \leq i \leq n}) \\ \beta &::= \text{assign}(\sigma). \psi & \psi &::= \neg \psi \mid \psi \vee \psi \mid \mathbf{X} \psi \mid \psi \mathbf{U} \psi \mid p \end{aligned}$$

where F^n ranges over semi-stable subsets of $\{0, 1\}^n$, for all $n \in \mathbb{N}$. The semantics of the operator F^n is defined as

$$\mathcal{G}, q \models_{\chi} F^n((\beta_i)_{i \leq n}) \iff \exists f \in F^n. \forall 1 \leq i \leq n. (f(i) = 1 \text{ iff } \mathcal{G}, q \models_{\chi} \beta_i).$$

Notice that if F^n would range over all subsets of $\{0, 1\}^n$, then this definition would exactly correspond to $\text{SL}[\text{BG}]^b$. Similarly, the case where $F^n = \{\mathbf{1}^n\}$ corresponds to $\text{SL}[\text{CG}]^b$, while $F^n = \{0, 1\}^n \setminus \{\mathbf{0}^n\}$ gives rise to $\text{SL}[\text{DG}]^b$.

► **Example 4.** Consider the following formula, expressing the existence of a Nash equilibrium for two players with respective LTL objectives ψ_1 and ψ_2 :

$$\exists x_1. \exists x_2. \forall y_1. \forall y_2. \bigwedge \left\{ \begin{array}{l} (\text{assign}(A_1 \mapsto y_1; A_2 \mapsto x_2). \psi_1) \Rightarrow (\text{assign}(A_1 \mapsto x_1; A_2 \mapsto x_2). \psi_1) \\ (\text{assign}(A_1 \mapsto x_1; A_2 \mapsto y_2). \psi_2) \Rightarrow (\text{assign}(A_1 \mapsto x_1; A_2 \mapsto x_2). \psi_2) \end{array} \right.$$

This formula has four goals, and it corresponds to the set

$$F^4 = \{(a, b, c, d) \in \{0, 1\}^4 \mid a \leq b \text{ and } c \leq d\}$$

Taking $f = (1, 1, 0, 0)$ and $g = (0, 0, 1, 1)$, with $s = (1, 0, 1, 0)$ we have $(f \wedge s) \vee (g \wedge \bar{s}) = (1, 0, 0, 1)$ and $(g \wedge s) \vee (f \wedge \bar{s}) = (0, 1, 1, 0)$, none of which is in F^4 . Hence our formula is not (syntactically) in $\text{SL}[\text{EG}]^b$.

The definition of $\text{SL}[\text{EG}]$ may look artificial. The main reason why we work with $\text{SL}[\text{EG}]$ is that it is maximal for the first claim of Theorem 8 (see Prop. 11). But as the next result shows, it is actually a large fragment encompassing $\text{SL}[\text{AG}]$ (hence also $\text{SL}[\text{CG}]$ and $\text{SL}[\text{DG}]$):

► **Proposition 9.** $\text{SL}[\text{EG}]^b$ contains $\text{SL}[\text{AG}]^b$. The inclusion is strict (syntactically).

5.2 Defining quasi-orders from semi-stable sets.

For $F^n \subseteq \{0, 1\}^n$, we write $\overline{F^n}$ for the complement of F^n . Fix such a set F^n , and pick $s \in \{0, 1\}^n$. For any $h \in \{0, 1\}^n$, we define

$$\begin{aligned} \mathbb{F}^n(h, s) &= \{h' \in \{0, 1\}^n \mid (h \wedge s) \vee (h' \wedge \bar{s}) \in F^n\} \\ \overline{\mathbb{F}^n}(h, s) &= \{h' \in \{0, 1\}^n \mid (h \wedge s) \vee (h' \wedge \bar{s}) \in \overline{F^n}\} \end{aligned}$$

Trivially $\mathbb{F}^n(h, s) \cap \overline{\mathbb{F}^n}(h, s) = \emptyset$ and $\mathbb{F}^n(h, s) \cup \overline{\mathbb{F}^n}(h, s) = \{0, 1\}^n$. If we assume F^n to be semi-stable, then the family $(\mathbb{F}^n(h, s))_{h \in \{0, 1\}^n}$ enjoys the following property:

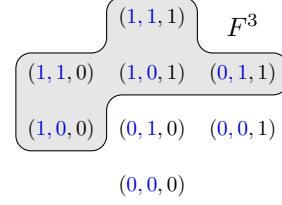
► **Lemma 10.** Fix a semi-stable set F^n and $s \in \{0, 1\}^n$. For any $h_1, h_2 \in \{0, 1\}^n$, either $\mathbb{F}^n(h_1, s) \subseteq \mathbb{F}^n(h_2, s)$ or $\mathbb{F}^n(h_2, s) \subseteq \mathbb{F}^n(h_1, s)$.

Given a semi-stable set F^n and $s \in \{0, 1\}^n$, we can use the inclusion relation of Lemma 10 to define a relation $\preceq_s^{F^n}$ (written \preceq_s when F^n is clear) over the elements of $\{0, 1\}^n$. It is defined as follows: $h_1 \preceq_s h_2$ if, and only if, $\mathbb{F}^n(h_1, s) \subseteq \mathbb{F}^n(h_2, s)$.

This relation is a quasi-order: its reflexivity and transitivity both follow from the reflexivity and transitivity of the inclusion relation \subseteq . By Lemma 10, this quasi-order is total. Intuitively, \preceq_s orders the elements of $\{0, 1\}^n$ based on how “easy” it is to complete

their restriction to s so that the completion belongs to F^n . In particular, only the indices on which s take value 1 are used to check whether $h_1 \preceq_s h_2$: given $h_1, h_2 \in \{0, 1\}^n$ such that $(h_1 \wedge s) = (h_2 \wedge s)$, we have $\mathbb{F}(h_1, s) = \mathbb{F}(h_2, s)$, and $h_1 \equiv_s h_2$.

► **Example 5.** Consider the set $F^3 = \{(1, 0, 0), (1, 1, 0), (1, 0, 1), (0, 1, 1), (1, 1, 1)\}$ represented on the figure opposite, and which can be shown to be semi-stable. Fix $s = (1, 1, 0)$. Then $\mathbb{F}^3((0, 1, \star), s) = \{0, 1\}^2 \times \{1\}$, while $\mathbb{F}^3((1, 1, \star), s) = \mathbb{F}^3((1, 0, \star), s) = \{0, 1\}^3$ and $\mathbb{F}^3((0, 0, \star), s) = \emptyset$. It follows that $(0, 0, \star) \preceq_s (0, 1, \star) \preceq_s (1, 0, \star) \equiv_s (1, 1, \star)$.



5.3 Sketch of proof of Theorem 8

The approach we used in Prop 2 does not extend in general to formulas with several goals. Consider for instance formula $(Q_i x_i)_{i \leq l} (\beta_1 \cdot \psi_1 \Leftrightarrow \beta_2 \cdot \psi_2)$: if at some points the two goals give rise to two different outcomes (hence to two different subgames), the winning objectives in one subgame depends on what is achieved in the other subgame.

$\text{SL}[\text{EG}]^b$ has been designed to prevent such situations: when two (or more) outcomes are available at a given position, each *subgame* can be assigned an independent winning objective. This objective can be obtained from the quasi-orders \preceq_s associated with the $\text{SL}[\text{EG}]^b$ formula being considered. Consider again Example 5: associating the set F^3 with three goals β_1 , β_2 and β_3 , we get a formula in $\text{SL}[\text{EG}]^b$. Assume that the moves selected by the players give rise to the same transition for β_1 and β_2 , and to a different transition for β_3 ; then in the subgame reached when following the transition of β_1 and β_2 (hence with $s = (1, 1, 0)$), the optimal way of fulfilling goals β_1 and β_2 is given by $(0, 0, \star) \preceq_s (0, 1, \star) \preceq_s (1, 0, \star) \equiv_s (1, 1, \star)$, independently of what may happen in the subgame reached by following the transition given by β_3 .

We exploit this idea in our proof: first, in order to keep track of the truth values of the LTL formulas ψ_i of each goal, we define a family of parity automata, one for each subset of goals of the formula under scrutiny. A subgame, as considered above, is characterized by a state q of the original concurrent game, a state d_p of each of the parity automata, and a vector $s \in \{0, 1\}^n$ defining which goals are still active. For each subgame, we can compute, by induction on s , the optimal set of goals that can be fulfilled from that configuration. The optimal strategies of both players in each subgame can be used to define (partial) optimal timeline dependence maps. We can then combine these partial maps together to get optimal dependence maps θ and $\bar{\theta}$; using similar arguments as for the proof of Prop. 6, we get a valuation χ such that $\theta(\chi|_{\forall}) = \chi = \bar{\theta}(\chi|_{\exists})$, from which we deduce that exactly one of φ and $\neg\varphi$ holds.

5.4 Maximality of $\text{SL}[\text{EG}]^b$

Finally, we prove that $\text{SL}[\text{EG}]^b$ is, in a sense, maximal for the first property of Theorem 8:

► **Proposition 11.** For any non-semi-stable boolean set $F^n \subseteq \{0, 1\}^n$, there exists a $\text{SL}[\text{BG}]^b$ formula φ built on F^n , a game \mathcal{G} and a state q_0 such that $\mathcal{G}, q_0 \not\models^T \neg\varphi$ and $\mathcal{G}, q_0 \models^T \varphi$.

Whether $\text{SL}[\text{EG}]^b$ is also maximal for having a 2-EXPTIME model-checking algorithm remains open. Actually, we do not know if $\text{SL}[\text{BG}]^b$ model checking is decidable under the timeline semantics. These questions will be part of our future works on this topic.

References

- 1 Thomas Ágotnes, Valentin Goranko, and Wojciech Jamroga. Alternating-time temporal logics with irrevocable strategies. In Dov Samet, editor, *Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge (TARK'07)*, pages 15–24, June 2007.
- 2 Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, September 2002. doi:10.1145/585265.585270.
- 3 Patricia Bouyer, Patrick Gardy, and Nicolas Markey. Weighted strategy logic with boolean goals over one-counter games. In Prahladh Harsha and G. Ramalingam, editors, *Proceedings of the 35th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'15)*, volume 45 of *Leibniz International Proceedings in Informatics*, pages 69–83. Leibniz-Zentrum für Informatik, December 2015. doi:10.4230/LIPIcs.FSTTCS.2015.69.
- 4 Patricia Bouyer, Patrick Gardy, and Nicolas Markey. On the semantics of strategy logic. *Information Processing Letters*, 116(2):75–79, February 2016. doi:10.1016/j.ipl.2015.10.004.
- 5 Romain Brenguier, Jean-François Raskin, and Ocan Sankur. Assume-admissible synthesis. *Acta Informatica*, 54(1):41–83, February 2017. doi:10.1007/s00236-016-0273-2.
- 6 Thomas Brihaye, Arnaud Da Costa, François Laroussinie, and Nicolas Markey. ATL with strategy contexts and bounded memory. In Sergei N. Artemov and Anil Nerode, editors, *Proceedings of the International Symposium Logical Foundations of Computer Science (LFCS'09)*, volume 5407 of *Lecture Notes in Computer Science*, pages 92–106. Springer-Verlag, January 2009. doi:10.1007/978-3-540-92687-0_7.
- 7 Krishnendu Chatterjee, Thomas A. Henzinger, and Nir Piterman. Strategy logic. In Luís Caires and Vasco T. Vasconcelos, editors, *Proceedings of the 18th International Conference on Concurrency Theory (CONCUR'07)*, volume 4703 of *Lecture Notes in Computer Science*, pages 59–73. Springer-Verlag, September 2007. doi:10.1007/978-3-540-74407-8_5.
- 8 Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In Dexter C. Kozen, editor, *Proceedings of the 3rd Workshop on Logics of Programs (LOP'81)*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer-Verlag, 1982. doi:10.1007/BFb0025774.
- 9 Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model checking*. MIT Press, 2000.
- 10 Rodica Condurache, Emmanuel Filiot, Raffaella Gentilini, and Jean-François Raskin. The complexity of rational synthesis. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP'16) – Part II*, volume 55 of *Leibniz International Proceedings in Informatics*, pages 121:1–121:15. Leibniz-Zentrum für Informatik, July 2016. doi:10.4230/LIPIcs.ICALP.2016.121.
- 11 E. Allen Emerson and Charanjit S. Jutla. Tree automata, mu-calculus and determinacy. In *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science (FOCS'91)*, pages 368–377. IEEE Comp. Soc. Press, October 1991. doi:10.1109/SFCS.1991.185392.
- 12 Dana Fisman, Orna Kupferman, and Yoad Lustig. Rational synthesis. In Javier Esparza and Rupak Majumdar, editors, *Proceedings of the 16th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'10)*, volume 6015 of *Lecture Notes in Computer Science*, pages 190–204. Springer-Verlag, March 2010. doi:10.1007/978-3-642-12002-2_16.
- 13 Valentin Goranko and Govert van Drimmelen. Complete axiomatization and decidability of alternating-time temporal logic. *Theoretical Computer Science*, 353(1-3):93–117, March 2006.

- 14 Orna Kupferman, Giuseppe Perelli, and Moshe Y. Vardi. Synthesis with rational environments. *Annals of Mathematics and Artificial Intelligence*, 78(1):3–20, September 2016. doi:10.1007/s10472-016-9508-8.
- 15 François Laroussinie and Nicolas Markey. Augmenting ATL with strategy contexts. *Information and Computation*, 245:98–123, December 2015. doi:10.1016/j.ic.2014.12.020.
- 16 François Laroussinie, Nicolas Markey, and Ghassan Oreiby. On the expressiveness and complexity of ATL. *Logical Methods in Computer Science*, 4(2), May 2008. doi:10.2168/LMCS-4(2:7)2008.
- 17 Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe Y. Vardi. What makes ATL* decidable? A decidable fragment of strategy logic. In Maciej Koutny and Irek Ulidowski, editors, *Proceedings of the 23rd International Conference on Concurrency Theory (CONCUR'12)*, volume 7454 of *Lecture Notes in Computer Science*, pages 193–208. Springer-Verlag, September 2012.
- 18 Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe Y. Vardi. Reasoning about strategies: On the model-checking problem. *ACM Transactions on Computational Logic*, 15(4):34:1–34:47, August 2014. doi:10.1145/2631917.
- 19 Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe Y. Vardi. Reasoning about strategies: On the satisfiability problem. *Logical Methods in Computer Science*, 13(1), March 2017. doi:10.23638/LMCS-13(1:9)2017.
- 20 Fabio Mogavero, Aniello Murano, and Luigi Sauro. On the boundary of behavioral strategies. In *Proceedings of the 28th Annual Symposium on Logic in Computer Science (LICS'13)*, pages 263–272. IEEE Comp. Soc. Press, June 2013.
- 21 Fabio Mogavero, Aniello Murano, and Luigi Sauro. A behavioral hierarchy of strategy logic. In Nils Bulling, Leendert W. N. van der Torre, Serena Villata, Wojciech Jamroga, and Wamberto Weber Vasconcelos, editors, *Proceedings of the 15th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA'14)*, volume 8624 of *Lecture Notes in Artificial Intelligence*, pages 148–165. Springer-Verlag, August 2014. doi:10.1007/978-3-319-09764-0_10.
- 22 Fabio Mogavero, Aniello Murano, and Moshe Y. Vardi. Reasoning about strategies. In Kamal Lodaya and Meena Mahajan, editors, *Proceedings of the 30th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'10)*, volume 8 of *Leibniz International Proceedings in Informatics*, pages 133–144. Leibniz-Zentrum für Informatik, December 2010. doi:10.4230/LIPIcs.FSTTCS.2010.133.
- 23 Andrzej Mostowski. Games with forbidden positions. Research Report 78, University of Danzig, 1991.
- 24 Amir Pnueli. The temporal logic of programs. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS'77)*, pages 46–57. IEEE Comp. Soc. Press, October–November 1977. doi:10.1109/SFCS.1977.32.
- 25 Jean-Pierre Queille and Joseph Sifakis. Specification and verification of concurrent systems in CESAR. In Mariangiola Dezani-Ciancaglini and Ugo Montanari, editors, *Proceedings of the 5th International Symposium on Programming (SOP'82)*, volume 137 of *Lecture Notes in Computer Science*, pages 337–351. Springer-Verlag, April 1982. doi:10.1007/3-540-11494-7_22.
- 26 Farn Wang, Chung-Hao Huang, and Fang Yu. A temporal logic for the interaction of strategies. In Joost-Pieter Katoen and Barbara König, editors, *Proceedings of the 22nd International Conference on Concurrency Theory (CONCUR'11)*, volume 6901 of *Lecture Notes in Computer Science*, pages 466–481. Springer-Verlag, September 2011.