Theory of Computing Systems (2020) 64:467–507
https://doi.org/10.1007/s00224-019-09926-y

Dependences in Strategy Logic

Patrick Gardy[1,2] · **Patricia Bouyer**[1] · **Nicolas Markey**[1,3]

Published online: 8 January 2020

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.

Keywords Temporal logic · Multiple-player games · Strategy · Model checking

1 Introduction

Temporal logics Since Pnueli's seminal paper [36] in 1977, temporal logics have been widely used in theoretical computer science, especially by the formal-verification

This article is part of the Topical Collection on *Special Issue on Theoretical Aspects of Computer Science (2018)*

This work was supported by ERC project EQualIS (StG-308087). A preliminary version of this paper appeared in the proceedings of STACS'18 [19].

✉ Nicolas Markey
nicolas.markey@irisa.fr

[1] LSV, CNRS, ENS Paris-Saclay, University of Paris-Saclay, Saint-Aubin, France

[2] Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China

[3] Irisa, University of Rennes & CNRS & Inria, Rennes, France

community. Temporal logics provide powerful languages for expressing properties of reactive systems, and enjoy efficient algorithms for satisfiability and model checking [13]. Since the early 2000s, new temporal logics have appeared to address *open* and *multi-agent systems*. While classical temporal logics (e.g. CTL [12, 37] and LTL [36]) 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. This can be used to specify that a controller can enforce safety of a whole system, whatever the other components do. This is usually seen as a game where the controller plays against the other components, with the aim of maintaining safety of the global system; ATL can then express the existence of a winning strategy in such a game. ATL has been extensively studied since its introduction, both about its expressiveness and about its verification algorithms [2, 20, 28].

Adding strategic interactions in temporal logics 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). Indeed, ATL expresses properties such as "*Player A has a strategy to enforce φ*" (denoted $\langle\langle A\rangle\rangle\varphi$), where φ is a property to be fulfilled along *any* execution resulting from the selected strategy; in other terms, this existential quantification over strategies of A always implicitly contains a universal quantification over all the strategies of all the other players. This only allows to express *zero-sum* objectives.

Over the last 10 years, various extensions have been defined and studied in order to allow for more strategy interactions [1, 8, 11, 30, 39]. *Strategy Logic* (SL for short) [11, 30] 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 Player B would be written as $\exists\sigma_A.\ \forall\sigma_B.\ \mathsf{assign}(A \mapsto \sigma_A;\ B \mapsto \sigma_B).\ \varphi_A$. This precisely corresponds to formula $\langle\langle A\rangle\rangle\ \varphi_A$ of ATL (if the game only has two players).

SL can express much more: for example, it can express the existence of a strategy for Player A which allows Player B to satisfy one of two goals φ_B or φ'_B: we would write

$$\exists\sigma_A.\ [(\exists\sigma_B.\ \mathsf{assign}(A\mapsto\sigma_A;\ B\mapsto\sigma_B).\ \varphi_B)\wedge(\exists\sigma'_B.\ \mathsf{assign}(A\mapsto\sigma_A;\ B\mapsto\sigma'_B).\ \varphi'_B).$$

This expresses *collaborative* properties which are out of reach of ATL: formula $\langle\langle A\rangle\rangle\ (\langle\langle B\rangle\rangle\ \varphi_B \wedge \langle\langle B\rangle\rangle\varphi'_B)$ in ATL is equivalent to $(\langle\langle B\rangle\rangle\varphi_B \wedge \langle\langle B\rangle\rangle\varphi'_B$, since $\langle\langle B\rangle\rangle\ \varphi_B$ is understood as the existence of a winning strategy against any strategy of the other player(s).

As a last example, SL can express classical concepts in game theory, such as Nash equilibria with Boolean objectives. This provides an easy way of showing decidability of rational synthesis [14, 18, 26] or assume-admissible synthesis [7]): for instance, the existence of an admissible strategy for objective φ of Player A (i.e., a strategy that is strictly dominated by no other strategies [7]) is expressed as

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

Such a formula shows that complex strategy interactions may be useful for expressing classical properties of multi-player games.

This series of examples illustrates how SL is both expressive and convenient, at the expense of a very high complexity: SL model checking has non-elementary complexity (and satisfiability is undecidable, unless the problem is restricted to turn-based game structures) [27, 30].

The high expressiveness of this logic, together with the decidability of its model-checking problem, has led to numerous studies around SL, either considering fragments of the logic with more efficient algorithms, or more expressive variants of the logic (e.g. with quantitative aspects), or variations on the notion of strategies (e.g. with limited observation of the game).

On the one hand, limitations have been imposed to strategic interactions in order to get more efficient algorithms [29, 32]. A *goal* is an LTL condition imposed to a strategy profile (built from quantified strategies). The fragment SL[1G] then contains formulas in prenex form with a single goal (and nested combinations thereof); this fragment is very close to ATL* [2] in terms of expressiveness, and its model-checking problem is 2-EXPTIME-complete. A BDD-based implementation of the model-checking algorithm for SL[1G], using a translation to parity games, is implemented in the tool MCMAS [10]. Several other fragments have been considered, e.g. allowing conjunctions (SL[CG]), disjunctions (SL[DG]), or general boolean combinations of goals (SL[BG]); model checking still is in 2-EXPTIME for the first two fragments [32], but it is non-elementary for SL[BG] [5].

On the other hand, various extensions have also been considered, in order to see how far the logic can be extended while preserving decidable model checking. In Graded SL, (existential) strategy quantifiers are decorated with quantitative constraints on the cardinality of the set of strategies satisfying a formula; this can be used e.g. to express uniqueness on Nash equilibria. Model checking is decidable (with non-elementary complexity) for Graded SL [3]. On a different note, Prompt SL extends SL with a parameterized modality $\mathbf{F}_{\leq n}\varphi$, which bounds the number of steps within which φ has to hold. Similarly, Bounded-Outcome SL adds a bound on the number of outcomes that must satisfy a given path formula. Again, model checking is decidable for those extensions [17].

Finally, SL has also been studied with different notions of strategies. When limiting strategy quantification to memoryless strategies, model checking is PSPACE-complete (as there are exponentially many strategies), but satisfiability is undecidable even for turn-based game structures [27]. Different types of strategies, based on sequences of actions, states or atomic propositions, are also considered in [22], with a focus on bisimulation invariance. When considering partial-observation strategies, model checking is undecidable (as is already the case for ATL [15]); a decidable fragment of SL is identified in [4], with a *hierarchical* restriction on nested strategy quantifiers. This study of imperfect-information games has been extended with epistemic variants of SL, which allows to reason about the knowledge of agents. Model checking is undecidable in the general case, but several papers identify specific settings where model checking is decidable [9, 21, 25].

Understanding SL 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: when selecting a strategy to be played later, are the intermediary events part of the *memory* of that strategy? While both options may make sense depending on the applications, only one of them makes model checking decidable [6].

A second phenomenon—which is the main focus of the present paper—concerns *strategy dependences* [30]: in a formula such as $\forall \sigma_A.\ \exists \sigma_B.\ \varphi$, 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 [30, 33], 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 [33], based on a kind of Skolemization of the formula: for instance, a formula of the form $(\forall x_i \exists y_i)_i.\ \varphi$ 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* [33] 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 on several fragments of SL (namely SL[1G], SL[CG] and SL[DG]), the classic and elementary semantics would coincide [29, 32].

The coincidence actually only holds for SL[1G]. As we explain in this paper, elementary dependences as defined and used in [29, 32] do not exactly capture the intuition that strategies should depend on the "behavior [of universal strategies] on the history of interest only" [32]: 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. 5).

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 [30, 33]. 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 2-EXPTIME-complete.

Related works To the best of our knowledge, strategy dependences have only been considered in a series of recent works by Mogavero et al. [29, 30, 32, 33], both as a way of making the semantics of SL more realistic in certain situations, and as a way of lowering the algorithmic complexity of verification of certain fragments of SL.

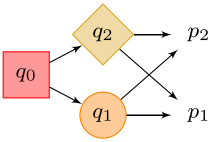
The question of the dependence of quantifiers in first-order logic is an old topic: in [23], *branching quantifiers* are introduced to define how quantified variables may depend on each other. Similarly, Dependence Logic [38] and Independence-Friendly Logic [24] also add such restrictions on dependences of quantified variables on top of first-order logic. While the settings are quite different to ours, the underlying ideas are similar, and in particular share an interpretation in terms of games of imperfect information.

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

Fix a state $q \in \mathsf{Q}$. 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 $q_i \in \mathsf{succ}(q_{i-1})$ for all $i > 0$. We write $\mathsf{Play}_{\mathcal{G}}(q)$ for the set of plays in $\mathcal{G}$ from q. In this and all similar notations, we might omit



$$\varphi = \forall y.\exists z.\forall x_A.\forall x_B.\ \bigvee \begin{cases} \mathsf{assign}(\blacksquare \mapsto x_A; \bullet \mapsto y; \blacklozenge \mapsto z).\ \mathbf{F}\,p_1 \\ \mathsf{assign}(\blacksquare \mapsto x_B; \bullet \mapsto y; \blacklozenge \mapsto z).\ \mathbf{F}\,p_2 \end{cases}$$

Fig. 1 A game and a SL[BG] formula

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 $\mathsf{Pref}(\pi)$ for the set of strict prefixes of play π. Such finite prefixes are called *histories*, and we let $\mathsf{Hist}_{\mathcal{G}}(q) = \mathsf{Pref}(\mathsf{Play}_{\mathcal{G}}(q))$. We extend the notion of strict prefixes and the notation Pref to histories in the natural way, requiring in particular that $\rho \notin \mathsf{Pref}(\rho)$. A (finite) extension of a history ρ is any history ρ' such that $\rho \in \mathsf{Pref}(\rho')$. Let $\rho = (q_i)_{i \leq L}$ be a history. We define $\mathsf{first}(\rho) = q_0$ and $\mathsf{last}(\rho) = q_L$. Let $\rho' = (q'_j)_{j \leq L'}$ be a history from $\mathsf{last}\rho$. 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 $\delta \colon \mathsf{Hist}_{\mathcal{G}}(q) \to \mathsf{Act}$. We write $\mathsf{Strat}_{\mathcal{G}}(q)$ for the set of strategies in $\mathcal{G}$ from q. Given a strategy $\delta \in \mathsf{Strat}(q)$ and a history ρ from q, the *translation* $\delta_{\overrightarrow{\rho}}$ of δ by ρ is the strategy $\delta_{\overrightarrow{\rho}}$ from $\mathsf{last}\rho$ defined by $\delta_{\overrightarrow{\rho}}(\rho') = \delta(\rho \cdot \rho')$ for any $\rho' \in \mathsf{Hist}(\mathsf{last}(\rho))$. A *context* (sometimes also called *valuation*) from q is a partial function $\chi : \mathcal{V} \cup \mathsf{Agt} \rightharpoonup \mathsf{Strat}(q)$. As usual, for any partial function f, we write $\mathsf{dom}(f)$ for the domain of f.

Let $q \in Q$ and χ be a context from q. If $\mathsf{Agt} \subseteq \mathsf{dom}(\chi)$, then χ induces a unique play from q, called its *outcome*, and defined as $\mathsf{out}(q, \chi) = (q_i)_{i \in \mathbb{N}}$ 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 $A \in \mathsf{Agt}$.

2.2 Strategy Logic with Boolean Goals

Strategy Logic (SL for short) was introduced in [11], and further extended and studied in [30, 34], 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 [31, 34] and model checking is TOWER-complete [5, 30]. In this paper, we focus on a restricted fragment of SL, called $\mathsf{SL[BG]}^\flat$ (where BG stands for *boolean goals* [30], and the symbol $\flat$ indicates that we do not allow nesting of (closed) subformulas; we discuss this latter restriction below).

Syntax Formulas in $\mathsf{SL[BG]}^\flat$ are built along the following grammar

$$\begin{array}{lll} \mathsf{SL[BG]}^\flat \ni \varphi ::= \exists x.\, \varphi \mid \forall x.\, \varphi \mid \xi & \quad & \xi ::= \neg\xi \mid \xi \wedge \xi \mid \xi \vee \xi \mid \omega \\ \qquad\qquad \omega ::= \mathsf{assign}(\sigma).\, \psi & & \psi ::= \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathbf{X}\psi \mid \psi \,\mathbf{U}\, \psi \mid p \end{array}$$

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

Free variables With any subformula ζ of some formula $\varphi \in \mathsf{SL[BG]}^\flat$, we associate its set of *free agents and variables*, which we write $\mathsf{free}(\zeta)$. 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 $\mathsf{SL[BG]}^\flat$ below). The set $\mathsf{free}\zeta$ is defined inductively:

$$\begin{array}{rclrcl}
\mathsf{free}(p) &=& \varnothing \quad \text{for all } p \in \mathsf{AP} & \mathsf{free}(\mathbf{X}\,\psi) &=& \mathsf{Agt} \cup \mathsf{free}(\psi) \\
\mathsf{free}(\neg\alpha) &=& \mathsf{free}(\alpha) & \mathsf{free}(\psi_1 \mathbf{U} \psi_2) &=& \mathsf{Agt} \cup \mathsf{free}(\psi_1) \cup \mathsf{free}(\psi_2) \\
\mathsf{free}(\alpha_1 \vee \alpha_2) &=& \mathsf{free}(\alpha_1) \cup \mathsf{free}(\alpha_2) & \mathsf{free}(\exists x.\, \varphi) &=& \mathsf{free}(\varphi) \setminus \{x\} \\
\mathsf{free}(\alpha_1 \wedge \alpha_2) &=& \mathsf{free}(\alpha_1) \cup \mathsf{free}(\alpha_2) & \mathsf{free}(\forall x.\, \varphi) &=& \mathsf{free}(\varphi) \setminus \{x\}
\end{array}$$

$$\mathsf{free}(\mathsf{assign}(\sigma).\, \varphi) = (\mathsf{free}(\varphi) \cup \sigma(\mathsf{Agt} \cap \mathsf{free}(\varphi))) \setminus \mathsf{Agt}$$

Subformula ζ is said to be *closed* whenever $\mathsf{free}(\zeta) = \varnothing$. We can now comment on our choice of considering the flat fragment of $\mathsf{SL[BG]}$: the full fragment, as defined in [30], allows for nesting *closed* $\mathsf{SL[BG]}$ formulas in place of atomic propositions. The meaning of such nesting in our setting is ambiguous, because our semantics (in Sections 3–5) 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 context $\chi \colon \mathcal{V} \cup \mathsf{Agt} \rightharpoonup \mathsf{Strat}(q)$. We inductively define the semantics of a subformula α of a formula of $\mathsf{SL[BG]}^\flat$ at q under context χ, requiring $\mathsf{free}(\alpha) \subseteq \mathsf{dom}(\chi)$. We omit the easy cases of boolean combinations and atomic propositions.

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

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

Notice that, writing $\chi' = \chi[A \in \mathsf{Agt} \mapsto \chi(\sigma(A))]$, we have $\mathsf{free}(\psi) \subseteq \mathsf{dom}(\chi')$ if $\mathsf{free}(\alpha) \subseteq \mathsf{dom}(\chi)$, 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 $\mathsf{Agt} \subseteq \mathsf{free}(\psi) \subseteq \mathsf{dom}(\chi)$, the context χ induces a unique outcome $\mathsf{out}(q, \chi) = (q_i)_{i \in \mathbb{N}}$ from q. For $n \in \mathbb{N}$, we write $\mathsf{out}_n(q, \chi) = (q_i)_{i \leq n}$, and define $\chi_{\overrightarrow{n}}$ as the context obtained by shifting all the strategies in the image of χ by $\mathsf{out}_n(q, \chi)$. Under the same conditions, we also define $q_{\overrightarrow{n}} = \mathsf{last}(\mathsf{out}_n(q, \chi))$. We then set

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

In the sequel, we use classical shorthands, such as $\top$ for $p \vee \neg p$ (for any $p \in \mathsf{AP}$), $\mathbf{F}\psi$ 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:

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

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 $\mathsf{SL[BG]}$ 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 🟠); because $\mathcal{G}$ is turn-based, we identify the actions of the owner of a state with the resulting target state, so that $\delta_y(q_0q_1)$ will be either p_1 or p_2. We then define strategy δ_z (to be played by 🔶) as $\delta_z(q_0q_2) = \delta_y(q_0q_1)$. Then clearly, for any strategy assigned to player 🟥, 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 [29, 32, 33], by adding further restrictions to boolean combinations in the grammar defining the syntax:

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

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

- $(Q_i x_i)_{i\le l}$ is a block of quantifications, with $\{x_i \mid 1 \le i \le l\} \subseteq \mathcal{V}$ and $Q_i \in \{\exists, \forall\}$, for every $1 \le i \le l$;
- $\xi(g_1, ..., g_n)$ is a boolean combination of its arguments;
- for all $1 \le j \le n$, $\beta_j.\ \psi_j$ 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 [30, 33] and define the semantics of SL[BG]$^\flat$ 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 SL[BG]$^\flat$ formula $\varphi = (Q_i x_i)_{1\le i\le l}.\ \xi(\beta_j.\ \varphi_j)_{j\le n}$, assuming w.l.o.g. that $\{x_i \mid 1 \le i \le l\} = \mathcal{V}$. 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 $\theta\colon \mathsf{Strat}^{\mathcal{V}^\forall} \to \mathsf{Strat}^{\mathcal{V}}$ 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 $w \in \mathsf{Strat}^{\mathcal{V}^\forall}$, 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 $\mathsf{SL[BG]}^\flat$: generally speaking, formula $\varphi = (Q_i x_i)_{1\le i\le l}.\ \xi(\beta_j.\ \varphi_j)_{j\le n}$ holds true if there exists a φ-map θ such that, for any $w\colon \mathcal{V}^\forall \to \mathsf{Strat}$, the valuation $\theta(w)$ makes $\xi(\beta_j.\ \varphi_j)_{j\le n}$ hold true.

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

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

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 [29, 30] 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:

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

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).

Indeed, consider a map θ satisfying (E), and pick two strategy valuations w_1 and w_2 and an existential variable x_i such that

$$\forall x_j \in \mathcal{V}^\forall \cap \{x_j \mid j < i\}.\ w_1(x_j) = w_2(x_j).$$

In particular, for those x_j, we have $w_1(x_j)(\rho) = w_2(x_j)(\rho)$ for any history ρ (hence also for any of its prefixes). By (E), it follows $\theta(w_1)(x_i)(\rho) = \theta(w_2)(x_i)(\rho)$. Since this holds for any history, we have shown $\theta(w_1)(x_i) = \theta(w_2)(x_i)$.

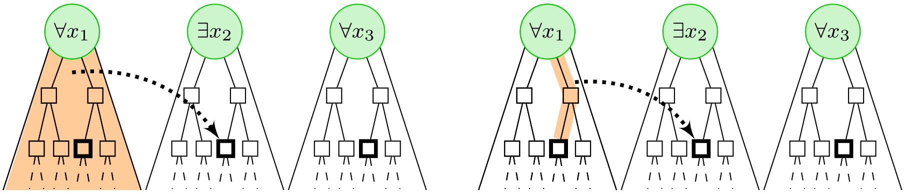


Fig. 2 Classical (left) vs elementary (right) dependences for a formula $\forall x_1.\ \exists x_2.\ \forall x_3.\ \xi$

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

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

As explained above, this actually corresponds to the usual semantics of SL[BG]$^\flat$ as given in Section 2 [30, 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* [30] as:

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

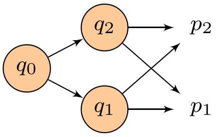
Again, when such a map exists, it is called an E-witness. Notice that since Property (E) implies Property (C), we have $\mathcal{G}, q \models^E \varphi \Rightarrow \mathcal{G}, q \models^C \varphi$ for any $\varphi \in$ SL[BG]$^\flat$. This corresponds to the intuition that it is harder to satisfy a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

The syntactic vs. semantic negations If $\varphi = (Q_i x_i)_{1\le i\le l}\xi(\beta_j.\ \varphi_j)_{j\le n}$ is an SL[BG]$^\flat$ formula, its syntactic negation $\neg\varphi$ is the formula $(\overline{Q}_i x_i)_{i\le l}(\neg\xi)(\beta_j.\ \varphi_j)_{j\le n}$, 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 classical semantics of Sl, 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 SL[1G]$^\flat$, but may fail to hold for SL[BG]$^\flat$.

Proposition 1 *There exist a game $\mathcal{G}$ with initial state q_0 and a formula $\varphi \in$ SL[BG]$^\flat$ 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 q_1 from q_0.

We abbreviate this as $\theta(w)(y)(q_0) = q_1$ in the sequel. Now, consider two valuations w_1 and w_2 such that $w_1(x)(q_0) = w_2(x)(q_0) = q_2$ and $w_1(x)(q_0 \cdot q_1) = w_2(x)(q_0 \cdot q_1)$, but such that $w_1(x)(q_0 \cdot q_2) = p_1$ and $w_2(x)(q_0 \cdot q_2) = p_2$. In order to fulfill the second goal under both valuations w_1 and w_2, we must have



$$\varphi = \forall x.\exists y.\ \bigwedge \begin{cases} \mathsf{assign}(\bullet \mapsto y).\ \mathbf{F}\, q_1 \\ \mathsf{assign}(\bullet \mapsto x).\ \mathbf{F}\, p_1 \Leftrightarrow \mathsf{assign}(\bullet \mapsto y).\ \mathbf{F}\, p_1 \end{cases}$$

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

$\theta(w_1)(y)(q_0 \cdot q_1) = p_1$ and $\theta(w_2)(y)(q_0 \cdot q_1) = p_2$. But this violates Property (E): since $w_1(x)$ and $w_2(x)$ coincide on q_0 and on $q_0 \cdot q_1$, we must have $\theta(w_1)(y)(q_0 \cdot q_1) = \theta(w_2)(y)(q_0 \cdot q_1)$.

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 $\theta(w)(y)(q_0) = q_1$ and $\theta(w)(y)(q_0 \cdot q_1) = w(x)(q_0 \cdot q_2)$ if $w(x)(q_0) = q_2$, and $\theta(w)(y)(q_0 \cdot q_1) = w(x)(q_0 \cdot q_1)$ if $w(x)(q_0) = q_1$. As explained above, this entails $\mathcal{G}, q_0 \not\models^C \neg\varphi$, and $\mathcal{G}, q_0 \not\models^E \neg\varphi$.

The proof above uses only one player and two quantifiers, but a complex combination of goals. The game and formula of Fig. 1 provide an alternative proof, with three players and four quantifiers, but a formula in $\mathsf{SL[DG]}^\flat$ (which also entails the result for $\mathsf{SL[CG]}^\flat$).

Indeed, we already proved (see Example 1) that $\mathcal{G}, q_0 \models^C \varphi$, by making strategy z play in q_2 in the same direction as what strategy y plays in q_1. Then it cannot be $\mathcal{G}, q_0 \models^E \neg\varphi$, since this would imply $\mathcal{G}, q_0 \models^C \neg\varphi$, and both φ and $\neg\varphi$ would hold, which is impossible in the classical semantics. Thus $\mathcal{G}, q_0 \not\models^E \neg\varphi$.

Now, in the elementary semantics, we require the existence of a dependence map θ, defining in particular $\theta(w)(z)(q_0 \cdot q_2)$, and such that $\theta(w)(z)(q_0 \cdot q_2) = \theta(w')(z)(q_0 \cdot q_2)$ whenever $w(y)(q_0) = w'(y)(q_0)$. Consider the following two valuations w and w':

$$\begin{array}{llll} w(y)(q_0) = q_1 & w(y)(q_0q_1) = p_1 & w(x_A)(q_0) = q_2 & w(x_B)(q_0) = q_1 \\ w'(y)(q_0) = q_1 & w'(y)(q_0q_1) = p_2 & w(x_A)(q_0) = q_1 & w(x_B)(q_0) = q_2. \end{array}$$

Since $w(y)(q_0) = w'(y)(q_0)$, we must have $\theta(w)(z)(q_0 \cdot q_2) = \theta(w')(z)(q_0 \cdot q_2)$. Then

– if $\theta(w)(z)(q_0 \cdot q_2) = p_2$, then under the strategies prescribed by $\theta(w)$, both disjuncts in φ are false.
– otherwise, $\theta(w)(z)(q_0 \cdot q_2) = p_1$, and under the strategies prescribed by $\theta(w')$, again both disjuncts are false.

It follows that $\mathcal{G}, q_0 \not\models^E \varphi$. □

We now prove hat this phenomenon does not occur in $\mathsf{SL[1G]}$:

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

Notice that this result follows from [30, Corollary 4.21], which states that $\models^C$ and $\models^E$ coincide on $\mathsf{SL[1G]}$. However, since it is central to our approach, we develop a (new) full proof of this result.

Proof We begin with intuitive explanations before giving full details. We encode the satisfaction relation $\mathcal{G}, q_0 \models^E \varphi$ into a two-player turn-based parity game: the first player of the parity game will be in charge of selecting the existentially-quantified strategies, and her opponent will select the universally-quantified ones. This will be encoded by replacing each state of $\mathcal{G}$ with a tree-shaped module as depicted on Fig. 4. Following the strategy assignment of the $\mathsf{SL[1G]}$ formula φ, the strategies selected

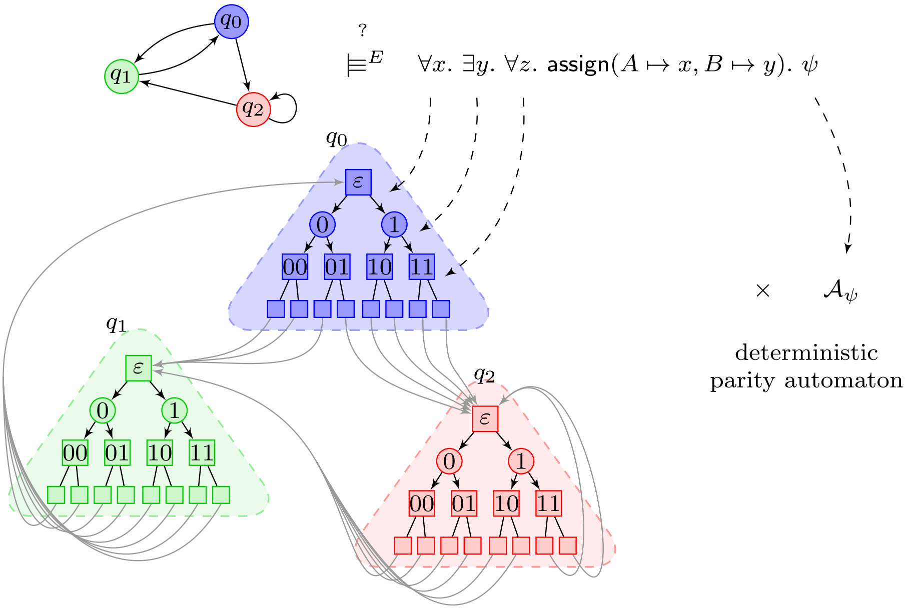

q_0
q_1
q_2
$\overset{?}{\models^E}$ $\forall x.\ \exists y.\ \forall z.$ assign$(A \mapsto x, B \mapsto y).\ \psi$
× $\mathcal{A}_\psi$
deterministic
parity automaton
ε
0
1
00
01
10
11

Fig. 4 Expressing $\mathcal{G}, q_0 \models^E \varphi$ as a two-player turn-based parity game

by those players will define a unique play, along which the LTL objective has to be fulfilled; this verification is encoded into a (doubly-exponential) parity automaton.

We prove that $\mathcal{G}, q_0 \models^E \varphi$ if, and only if, the first player wins; conversely, $\mathcal{G}, q_0 \not\models^E \varphi$ if the second player wins. Both claims crucially rely on the existence of memoryless optimal strategies for two-player parity games. Finally, by determinacy of those games, we get the expected result.

Notice that in this construction, Player $P_\exists$ has full observation, hence her moves may depend on all moves of Player $P_\forall$ along the current history. As a result, in our encoding, existentally-quantified strategies may depend on the value of *all* universally-quantified strategies along the current history; in the example of Fig. 4, this means that the moves selected by Player $P_\exists$ for x_1 may depend on the moves selected by Player $P_\forall$ for x_2 earlier in the game.

However, memoryless strategies are sufficient for both players to win parity games; a memoryles strategy for Player $P_\exists$ then precisely corresponds to an elementary dependence map, which proves our result.

We now give a full proof following this intuition.

Building a turn-based parity game $\mathcal{H}$ from $\mathcal{G}$ and φ For the rest of the proof, we fix a game $\mathcal{G}$ and a SL[1G] formula $\varphi = (Q_i x_i)_{i \leq l} \beta.\ \varphi$. Each state of $\mathcal{G}$ is replaced with a copy of the tree-shaped *quantification game* depicted on Fig. 4. A quantification game $\mathcal{Q}_\varphi$ is formally defined as follows:

– it involves two players, $P_\exists$ and $P_\forall$;

- the set of states is $S_\varphi = \{\mathfrak{m} \in \mathsf{Act}^* \mid 0 \leq |\mathfrak{m}| \leq l\}$, thereby defining a tree of depth $l+1$ with directions Act. A state $\mathfrak{m}$ in S_φ with $0 \leq |\mathfrak{m}| < l$ belongs to Player $P_\exists$ if, and only if, $Q_{|\mathfrak{m}|+1} = \exists$.
- from each $\mathfrak{m}$ with $0 \leq |\mathfrak{m}| < l$, for all $a \in \mathsf{Act}$, there is a transition to $\mathfrak{m} \cdot a$. The empty word $\epsilon \in S_\varphi$ is the starting node of the quantification game, and currently has no incoming transitions; states with $|\mathfrak{m}| = l$ also currently have no outgoing transitions.

A leaf (i.e., a state $\mathfrak{m}$ with $|\mathfrak{m}| = l$) in a quantification game represents a move vector of domain $\mathcal{V} = \{x_i \mid 1 \leq i \leq l\}$: we identify each leaf $\mathfrak{m}$ with the move vector $\mathfrak{m}$, hence writing $\mathfrak{m}(x_i)$ for $\mathfrak{m}(i)$.

We let D be a deterministic parity automaton over 2^{AP} associated with φ. We write d_0 for the initial state of D. Using quantification games, we can now define the turn-based parity game $\mathcal{H}$:

- it involves players $P_\exists$ and $P_\forall$;
- for each state q of $\mathcal{G}$ and each state d of D, $\mathcal{H}$ contains a copy of the quantification game $\mathcal{Q}_\varphi$, which we call the (q,d)*-copy*. Hence the set of states of $\mathcal{H}$ is the product of the state spaces of $\mathcal{G}$, D and $\mathcal{Q}_\varphi$.
- the transitions in $\mathcal{H}$ are of two types:
  - internal transitions in each copy of the quantification game are preserved;
  - consider a state $(q, d, \mathfrak{m})$ where $|\mathfrak{m}| = l$; this is a leaf in the quantification game. Let $q' = \Delta(q, m_\beta)$, where $m_\beta \colon \mathsf{Agt} \to \mathsf{Act}$ is the move vector over Agt defined by $m_\beta(A) = \mathfrak{m}(i-1)$ where $x_i = \beta(A)$ (i.e., assigning to each player $A \in \mathsf{Agt}$ the action $\mathfrak{m}(\beta(A))$); then we add a transition from $(q, d, \mathfrak{m})$ to (q', d', ϵ) where d' is the state of D reached from d when reading $\mathsf{lab}(q')$. Notice that $(q, d, \mathfrak{m})$ then has at most one outgoing transition.
- the priorities are inherited from those in D: state $(q, d, \mathfrak{m})$ has the same priority as d.

Correspondence between $\mathcal{G}$ and $\mathcal{H}$ We begin with building a correspondence between the runs and strategies in $\mathcal{G}$ and those in $\mathcal{H}$. In a sense, each step of a history in $\mathcal{G}$ is split into several steps in $\mathcal{H}$; we thus refine the notion of history in $\mathcal{G}$ in order to establish our correspondence. □

Definition 1 A *lane* in $\mathcal{G}$ is a tuple (ρ, u, b, t) made of

- a history $\rho = (q_j)_{0 \leq j \leq a}$ (for some integer a);
- a function $u \colon \mathcal{V} \times \mathsf{Pref}(\rho) \to \mathsf{Act}$;
- an integer $b \in [0; l]$;
- a function $t \colon \{x_1, ..., x_b\} \to \mathsf{Act}$ (t is the empty function if $b = 0$);

and such that

$$\forall 0 \leq j < a. \quad \Delta(q_j, (m_j(\beta(A)))_{A \in \mathsf{Agt}}) = q_{j+1} \quad \text{with } m_j \colon \mathcal{V} \to \mathsf{Act} \qquad x \mapsto u(x, \rho_{\leq j}) \quad (1)$$

We can then build a one-to-one application $\mathfrak{G}_p$ between histories in $\mathcal{H}$ and lanes in $\mathcal{G}$. With a history π in $\mathcal{H}$, written

$$\pi = \left(\prod_{0 \leq j < a} \prod_{0 \leq i \leq l} (q_j, d_j, \mathfrak{m}_{j,i}) \right) \cdot \prod_{0 \leq i \leq b} (q_a, d_a, \mathfrak{m}_{a,i}),$$

having length $a \cdot (l + 1) + b + 1$ with $0 \leq b < l$, we associate a lane $\mathfrak{G}_p(\pi) = ((q_j)_{j \leq a}, u, b, t)$ with

$$\begin{array}{rlcrl} u \colon \mathcal{V} \times \mathsf{Pref}(\rho) \to & \mathsf{Act} & & t \colon \{x_1, ..., x_b\} \to & \mathsf{Act} \\ x_i, (q_j)_{j \leq c} \mapsto & \mathfrak{m}_{c,i} & (\forall c < a) & x_i \mapsto & \mathfrak{m}_{a,i} \end{array}$$

The resulting function $\mathfrak{G}_p$ is clearly injective (different histories will correspond to different lanes), but also surjective. To prove the latter statement, we build the inverse function $\mathfrak{H}_p$: for a lane $((q_j)_{j \leq a}, u, b, t)$, we set $\mathfrak{H}_p((q_j)_{j \leq a}, u, b, t) = \pi$ where π is the history in $\mathcal{H}$ of length $a \cdot (l + 1) + b + 1$ defined as

$$\pi = \prod_{0 \leq j < a} \prod_{0 \leq i \leq l} \big(q_j, d_j, u(x_i, (q_{j'})_{j' \leq j})\big) \cdot \prod_{0 \leq i \leq b} \big(q_a, d_a, t(x_i, (q_j)_{j \leq a})\big)$$

where d_j is the state of D reached on input $(q_k)_{0 \leq k \leq j-1}$.

Because of the coherence condition (1), $\mathfrak{H}_p((q_j)_{j \leq a}, u, i, t)$ is indeed a history in $\mathcal{H}$. From the definitions, one can easily check that

$$\mathfrak{H}_p(\mathfrak{G}_p(\pi)) = \pi$$

and deduce that $\mathfrak{H}_p$ is the inverse function of $\mathfrak{G}_p$; therefore

Lemma 1 *The application $\mathfrak{G}_p$ is a bijection between lanes of $\mathcal{G}$ and histories in $\mathcal{H}$, and $\mathfrak{H}_p$ is its inverse function.*

Extending the correspondence We can use $\mathfrak{G}_p$ to describe another correspondence $\mathfrak{G}$ between strategies for $P_\exists$ in $\mathcal{H}$ and maps in $\mathcal{G}$. Remember that a map in $\mathcal{G}$ is a function $\theta \colon (\mathsf{Hist}_\mathcal{G} \to \mathsf{Act})^{\mathcal{V}^\forall} \to (\mathsf{Hist}_\mathcal{G} \to \mathsf{Act})^{\mathcal{V}}$. Remember also that if $Q_j = \forall$, then $\theta(w)(x_i)(\rho) = w(x_i)(\rho)$, so that we only have to define the map for existentially-quantified variables.

Formally, the application $\mathfrak{G}$ takes as input a strategy δ for player $P_\exists$ in $\mathcal{H}$, and returns a map in $\mathcal{G}$. It will enjoy the following properties:

- for any finite outcome π of δ in $\mathcal{H}$ ending at the root of a quantification game, there exists a function w such that $\mathfrak{G}_p(\pi) = (\rho, u, 0, t_\varnothing)$ where ρ is the outcome of $\mathfrak{G}(\delta)(w)$ in $\mathcal{G}$ under the assignment defined by β;
- conversely, for any path ρ in $\mathcal{G}$ that is an outcome of $\mathfrak{G}(\delta)(w)$ for some w and under the assignment defined by β, then letting $u(x, \rho') = \mathfrak{G}(\delta)(w)(x)(\rho')$, we have that $(\rho, u, 0, t_\emptyset)$ is a lane in $\mathcal{G}$ and $\mathfrak{H}_p(\rho, u, 0, t_\emptyset)$ is an outcome of δ in $\mathcal{H}$ ending in the root of a quantification game.

We fix δ, and for all w, ρ and x_i, we define $\mathfrak{G}(\delta)(w)(x_i)(\rho)$ by a double induction, first on the length of the history ρ in $\mathcal{G}$, and second on the sequence of variables x_i. We prove the properties above alongside the definition.

- **Initial step:** we begin with the case where ρ is the single state q_0.

  We proceed by induction on existentially-quantified variables, merging the initialization step with the induction step as they are similar. Consider an existentially-quantified variable x_i in $\mathcal{V}$. Given $w\colon \mathcal{V}^\forall \times \mathsf{Pref}(\rho) \cup \{\rho\} \to \mathsf{Act}$, we define a function $t_{i,w}\colon [x_1; x_{i-1}] \to \mathsf{Act}$ such that $t_{i,w}(x) = w(x, q_0)$ for $x \in \mathcal{V}^\forall \cap [x_1; x_{i-1}]$, and $t_{i,w}(x) = \mathfrak{G}(\delta)(w)(x)(q_0)$ for $x \in \mathcal{V}^\exists \cap [x_1; x_{i-1}]$, assuming that they have been defined in the previous induction steps on variables. We can then create the lane $lane_{i,w} = (\epsilon, u_\emptyset, i-1, t)$ and define

  $$\mathfrak{G}(\delta)(w)(x_i)(q_0) = \delta(\mathfrak{H}_p(lane_{i,w}))$$

  Pick an outcome π of δ in $\mathcal{H}$ of length $l+2$, and write $\mathfrak{m}$ for its $l+1$-st state: it defines a valuation for the variables in $\mathcal{V}$, hence defining a move vector m_β under the assignment β in Act. By construction of $\mathcal{H}$, this outcome ends in the state (q_1, d_1, ϵ) where $q_1 = \Delta(q_0, m_\beta)$ and d_1 is the successor of the initial state d_0 of D when reading $\mathsf{lab}(q_1)$. We now prove that $q_0 \cdot q_1$ is the outcome of $\mathfrak{G}(\delta)(w)$ for some w. For this, we let $w(x_i) = \mathfrak{m}_i$ for all $x_i \in \mathcal{V}^\forall$. By construction, $\mathfrak{G}(\delta)(w)(x_j)(q_0)$ precisely corresponds to $\mathfrak{m}(j)$, for all $x_j \in \mathcal{V}^\exists$. In the end, under assignment β, $\mathfrak{G}(\delta)(w)$ precisely returns the move vector m_β, hence proving our result.

  The proof of the converse statement follows similar arguments: consider an outcome $\rho = q_0 \cdot q_1$ of $\mathfrak{G}(\delta)(w)$ for some w. The lane $(\rho, u, 0, t_\emptyset)$ defined with $u(x, q_0) = \mathfrak{G}(\delta)(w)(x)(q_0)$ then corresponds through $\mathfrak{H}_p$ to a play ending in (q_1, d_1, ϵ), and visiting the leaf $\mathfrak{m}$ defined as $\mathfrak{m}_i = u(x_i, q_0)$. By construction, this is an outcome of δ in $\mathcal{H}$.
- **induction step:** we consider a history ρ in $\mathcal{G}$, assuming we have already defined $\mathfrak{G}(\delta)(w)(x_i)(\rho')$ for all prefix ρ' of ρ, and for all w and all variable x_i. We now define $\mathfrak{G}(\delta)(w)(x_i)(\rho)$, by induction on the list of variables. Again, the initialization step is merged with the induction step as they rely on the same arguments.

  Consider an existentially-quantified variable x_i, and $w\colon \mathcal{V}^\forall \times \mathsf{Pref}(\rho) \cup \{\rho\} \to \mathsf{Act}$. We define a function $t_{i,w}\colon [x_1; x_{i-1}] \to \mathsf{Act}$ where $t_{i,w}$ associate with $x \in \mathcal{V}^\forall \cap [x_1; x_{i-1}]$ the action $w(x)(\pi)$, and with $x \in \mathcal{V}^\exists \cap [x_1; x_{i-1}]$ the action $\mathfrak{G}(\delta)(w)(x)(\rho)$. We also define $u_w\colon \mathcal{V} \times \mathsf{Pref}(\rho) \to \mathsf{Act}$ as $u_w(x, \rho') = \mathfrak{G}(\delta)(w)(x)(\rho')$, for all prefixes ρ' of ρ. We can then create the lane $lane_{i,w} = (\pi, u_w, i-1, t_{i,w})$ and finally define

  $$\mathfrak{G}(\delta)(w)(x_i)(\rho) = \delta(\mathfrak{H}_p(lane_{i,w})).$$

  Using the same arguments as in the initial step, we prove our correspondence between the outcomes of δ in $\mathcal{H}$ and the outcomes of $\mathfrak{G}(\delta)$ in $\mathcal{G}$.

Notice that in the construction above, $\mathfrak{G}(\delta)(w)(x_i)(\rho)$ may depend on the value of $w(x_j, \rho')$ for $j > i$ and $\rho' \in \mathsf{Pref}(\rho)$: indeed, in the inductive definition, we define

$\mathfrak{S}(\delta)(w)(x_j)(\rho')$ before defining $\mathfrak{S}(\delta)(w)(x_i)(\rho)$. Hence in general $\mathfrak{S}(\delta)$ is *not* an elementary map.

However, in case δ is memoryless, we notice that $\mathfrak{S}(\delta)(w)(x_i)(\rho)$ only depends on value of δ in the last state of the lane $lane_{i,w}$, hence in particular not on u_w. This removes the above dependence, and makes $\mathfrak{S}(\delta)$ elementary.

Finally, notice that we can define a dual correspondence $\overline{\mathfrak{S}}$ relating strategies of Player $P_\forall$ and elementary maps in $\mathcal{G}$ where existential and universal variables are swapped.

Concluding the proof Using $\mathfrak{S}$, we prove our final correspondence between $\mathcal{H}$ and $\mathcal{G}$:

Lemma 2 *Assume that $P_\exists$ is winning in $\mathcal{H}$ and let δ be a positional winning strategy. Then the elementary map $\mathfrak{S}(\delta)$ is a witness that $\mathcal{G}, q_0 \models^E \varphi$.*

Similarly, assume that $P_\forall$ is winning in $\mathcal{H}$ and let $\overline{\delta}$ be a positional winning strategy. Then the elementary map $\overline{\mathfrak{S}}(\overline{\delta})$ is a witness that $\mathcal{G}, q_0 \models^E \neg\varphi$.

Proof We prove the first point, the second one following similar arguments. Assume that $P_\exists$ is winning in $\mathcal{H}$, and pick a memoryless winning strategy δ. Toward a contradiction, assume further that $\mathfrak{S}(\delta)$ is not a witness of $\mathcal{G}, q_0 \models^E \varphi$. Then there exists $w_0 \colon \mathcal{V}^\forall \to (\mathsf{Hist}_\mathcal{G} \to \mathsf{Act})$ s.t. $\mathcal{G}, q_0 \not\models_{\mathfrak{S}(\delta)(w_0)} \beta.\ \varphi$. We use w_0 to build a strategy $\overline{\delta}$ for Player $P_\forall$ in $\mathcal{H}$. Given a history

$$\pi = \prod_{0\le j<a} \prod_{0\le i\le l} (q_j, d_j, \mathfrak{m}_{j,i}) \cdot \prod_{0\le i\le b} (q_a, d_a, \mathfrak{m}_{a,i})$$

in $\mathcal{H}$, we define $\rho = \prod_{0\le j\le a} q_j$ and set $\overline{\delta}(\pi) = \mathfrak{S}(\delta)(w)(x_b)(\eta)$ where

- $w \colon \mathsf{Pref}(\rho) \cup \{\rho\} \times (\mathcal{V}^\forall \cap [x_1; x_b]) \to \mathsf{Act}$ is such that $w(\rho', x_i)$ is the action to be played for going from $\pi_{\le|\rho'|\cdot(l+1)+i-1}$ to $\pi_{\le|\rho'|\cdot(l+1)+i}$ in $\mathcal{H}$;
- $\eta = \prod_{0\le j<a} \prod_{0\le i\le l}(q_j, d_j, \mathfrak{m}_{j,i}))$.

Write $\nu = (q_j)_{j\in\mathbb{N}}$ for the outcome of $\theta(w_0)$ under strategy assignment β in $\mathcal{G}$. Then, by construction of $\overline{\delta}$, the outcome of δ and $\overline{\delta}$ in $\mathcal{H}$ will visit the $(q_j, d_j)_{j\in\mathbb{N}}$-copies of the quantification game, where d_j is the state reached by reading $(q_{j'})_{j'\le j}$ in the deterministic automaton D. Now, since $\mathcal{G}, q_0 \not\models_{\mathfrak{S}(\delta)(w_0)} \beta.\ \varphi$, we get that ν does not satisfy φ and therefore the outcome of δ and $\overline{\delta}$ in $\mathcal{H}$ does not satisfy the parity condition. This is in contradiction with δ being the winning strategy of $P_\exists$, and proves that $\mathfrak{S}(\delta)$ must be a witness that $\mathcal{G}, q_0 \models^E \varphi$.

Proposition 2, together with the determinacy of parity games [16, 35] immediately imply that at least one of φ and $\neg\varphi$ must hold in $\mathcal{G}$ for $\models^E$. This concludes our proof. □

The following two results, already mentioned in [30], immediately follow: the first result uses the fact that $\mathcal{G}, q_0 \models^E \varphi$ implies $\mathcal{G}, q_0 \models^C \varphi$; the second one uses the two-player game built in the proof.

Corollary 1 *The relations $\models^E$ and $\models^C$ coincide over* SL[1G].

Corollary 2 *Model checking* SL[1G] *is* 2-EXPTIME*-complete (for both semantics).*

Remark 1 As an immediate corollary of (the proof of) Prop. 1, we have that the relations $\models^C$ and $\models^E$ differ on $\mathsf{SL[CG]}^\flat$ (as well as on $\mathsf{SL[DG]}^\flat$). This contradicts the claim in [32] that $\models^E$ and $\models^C$ would coincide on SL[CG] (and SL[DG]).

Indeed, in [32], the satisfaction relation for SL[DG] and SL[CG] is encoded into a two-player game in pretty much the same way as we did in the proof of Proposition 2 for SL[1G]. 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 $\mathsf{SL[1G]}^\flat$ 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 $\mathsf{SL[BG]}^\flat$.

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. 5. 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:

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

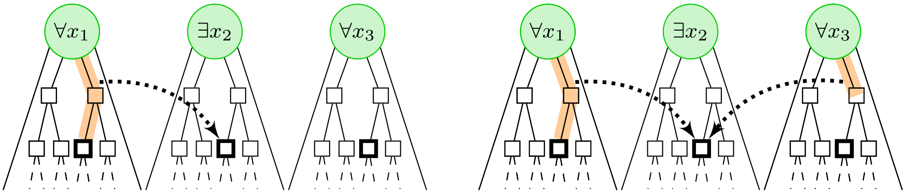


Fig. 5 Elementary (left) vs timeline (right) dependences for a formula $\forall x_1.\ \exists x_2.\ \forall x_3.\ \xi$

Using those maps, we introduce the *timeline semantics* of $\mathsf{SL[BG]}^\flat$:

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

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 $\varphi \in \mathsf{SL[BG]}^\flat$.

Example 2 Consider again the game of Fig. 1 in Section 2. We have seen that $\mathcal{G}, q_0 \models^C \varphi$ in Example 1, and that $\mathcal{G}, q_0 \not\models^E \varphi$ in the proof of Prop. 1. 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 3 *For any game $\mathcal{G}$ with initial state q_0, and any formula $\varphi \in \mathsf{SL[1G]}^\flat$, it holds $\mathcal{G}, q_0 \models^E \varphi \Leftrightarrow \mathcal{G}, q_0 \models^T \varphi$.*

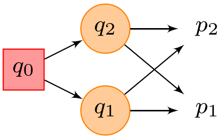
We now prove that this does not extend to $\mathsf{SL[CG]}^\flat$ and $\mathsf{SL[DG]}^\flat$:

Proposition 4 *The relations $\models^E$ and $\models^T$ differ on $\mathsf{SL[CG]}^\flat$, as well as on $\mathsf{SL[DG]}^\flat$.*

Proof The result for $\mathsf{SL[DG]}^\flat$ is witnessed by Example 2. For $\mathsf{SL[CG]}^\flat$, we consider the game structure and formula of Fig. 6. We first notice that $\mathcal{G}, q_0 \not\models^E \varphi$: 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 q_1 and q_2. But then no choice of x_B will make the second goal true.

On the other hand, considering the timeline semantics, strategy y after $q_0 \cdot q_1$ and $q_0 \cdot q_2$ may depend on the choice of x_A in q_0. When $w(x_A)(q_0) = q_1$, we let $\theta(w)(y)(q_0 \cdot q_1) = p_1$ and $\theta(w)(y)(q_0 \cdot q_2) = p_2$ and $\theta(w)(x_B)(q_0) = q_2$, which makes both goals hold true. Conversely, if $w(x_A)(q_0) = q_2$, then we let $\theta(w)(y)(q_0 \cdot q_2) = p_1$ and $\theta(w)(y)(q_0 \cdot q_1) = p_2$ and $\theta(w)(x_B)(q_0) = q_1$, which also defines a timeline map witnessing $\mathcal{G}, q_0 \not\models^E \varphi$. □

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



$$\varphi = \exists y.\ \forall x_A.\ \exists x_B.\ \bigwedge \begin{cases} \mathsf{assign}(\bigcirc \mapsto y;\ \square \mapsto x_A).\ \mathbf{F}\, p_1 \\ \mathsf{assign}(\bigcirc \mapsto y;\ \square \mapsto x_B).\ \mathbf{F}\, p_2 \end{cases}$$

Fig. 6 $\models^E$ and $\models^T$ differ on $\mathsf{SL[CG]}^\flat$

Prop. 3 and 2, the two negations coincide on $\mathsf{SL[1G]}^\flat$ under the timeline semantics. Moreover:

Proposition 5 *For any formula φ in $\mathsf{SL[BG]}^\flat$, for any game $\mathcal{G}$ and any state q_0, we have $\mathcal{G}, q_0 \models^T \varphi \Rightarrow \mathcal{G}, q_0 \not\models^T \neg\varphi$.*

Remember that the same result for $\models^E$ was proven easily from the implication $\mathcal{G}, q_0 \models^E \varphi \Rightarrow \mathcal{G}, q_0 \models^C \varphi$, and because the two negations coincide for $\models^C$. The proof for $\models^T$ is more involved.

Proof For a contradiction, assume that there exist two maps θ and $\overline{\theta}$ witnessing $\mathcal{G}, q_0 \models^T \varphi$ and $\mathcal{G}, q_0 \models^T \neg\varphi$ resp. Then

$$\forall w \colon \mathcal{V}^\forall \to (\mathsf{Hist} \to \mathsf{Act}). \qquad \mathcal{G}, q_0 \models_{\theta(w)} \xi(\beta_j.\, \varphi_j)_{j \le n} \tag{2}$$

$$\forall \overline{w} \colon \mathcal{V}^\exists \to (\mathsf{Hist} \to \mathsf{Act}). \qquad \mathcal{G}, q_0 \models_{\overline{\theta}(\overline{w})} \neg\xi(\beta_j.\, \varphi_j)_{j \le n} \tag{3}$$

From θ and $\overline{\theta}$, we build a strategy valuation χ on $\mathcal{V}$ such that $\theta(\chi_{|\mathcal{V}^\forall}) = \overline{\theta}(\chi_{|\mathcal{V}^\exists}) = \chi$. By (2) and (3), we get that $\mathcal{G}, q_0 \models_\chi \xi(\beta_j.\, \varphi_j)_{j \le n}$ and $\mathcal{G}, q_0 \models_\chi \neg\xi(\beta_j.\, \varphi_j)_{j \le n}$. It follows that there must exist a goal $\beta_j.\, \varphi_j$ for which $\mathcal{G}, q_0 \models_\chi \beta_j.\varphi_j$ and $\mathcal{G}, q_0 \models_\chi \neg\beta_j.\varphi_j$; then the outcome corresponding to β_j would satisfy both φ_j and $\neg\varphi_j$, which for LTL formulas is impossible.

We define $\chi(x)(\rho)$ inductively on histories and on the list of quantified variables. When ρ is the empty history q_0, we consider two cases:

- if $x_1 \in \mathcal{V}^\forall$, then $\overline{\theta}(\overline{w})(x_1)(q_0)$ does not depend on $\overline{w}$ at all, since $\overline{\theta}$ is a timeline-map. Hence we let $\chi(x_1)(q_0) = \overline{\theta}(\overline{w})(x_1)(q_0)$, for any $\overline{w}$.
- similarly, if $x_1 \in \mathcal{V}^\exists$, we let $\chi(x_1)(q_0) = \theta(w)(x_1)(q_0)$, which again does not depend on w.

Similarly, when $\chi(x)(q_0)$ has been defined for all $x \in \{x_1, ..., x_{i-1}\}$, we again consider two cases:

- if $x_i \in \mathcal{V}^\forall$, we define $\overline{w}(x_j)(q_0) = \chi(x_j)(q_0)$ for all $x_j \in \mathcal{V}^\exists \cap \{x_1, ..., x_{i-1}\}$, and let $\chi(x_i)(q_0) = \overline{\theta}(\overline{w})(x_i)(q_0)$, which again does not depend on the value of $\overline{w}$ besides those defined above;
- symmetrically, if $x_i \in \mathcal{V}^\exists$, we define $w(x_j)(q_0) = \chi(x_j)(q_0)$ for all $x_j \in \mathcal{V}^\forall \cap \{x_1, ..., x_{i-1}\}$, and let $\chi(x_i)(q_0) = \theta(w)(x_i)(q_0)$.

Notice that this indeed enforces that $\theta(\chi_{|\mathcal{V}^\forall})(x_i)(q_0) = \chi(x_i)(q_0)$ when $x_i \in \mathcal{V}^\exists$, and $\overline{\theta}(\chi_{|\mathcal{V}^\exists})(x_i)(q_0) = \chi(x_i)(q_0)$ when $x_i \in \mathcal{V}^\forall$.

The induction step is proven similarly: consider a history ρ and a variable x_i, assuming that χ has been defined for all variables on all prefixes of ρ, and for variables in $\{x_1, ..., x_{i-1}\}$ on ρ itself. Then:

- if $x_i \in \mathcal{V}^\forall$, we define $\overline{w}(x_j)(\rho') = \chi(x_j)(\rho')$ for all $x_j \in \mathcal{V}$ and all $\rho' \in \mathsf{Pref}(\rho)$, and $\overline{w}(x_j)(\rho) = \chi(x_j)(\rho)$ for all $x_j \in \mathcal{V}^\exists \cap \{x_1, ..., x_{i-1}\}$. We then let $\chi(x_i)(\rho) = \overline{\theta}(\overline{w})(x_i)(q_0)$, which does not depend on the value of $\overline{w}$ besides those defined above;
- the construction for the case when $x_i \in \mathcal{V}^\exists$ is similar.

As in the initial step, it is easy to check that this construction enforces $\theta(\chi_{|\mathcal{V}^\forall}) = \overline{\theta}(\chi_{|\mathcal{V}^\exists}) = \chi$, as required. □

Proposition 6 *There exists a formula $\varphi \in$ SL[BG]$^\flat$, a (turn-based) game $\mathcal{G}$ and a state q_0 such that $\mathcal{G}, q_0 \not\models^T \varphi$ and $\mathcal{G}, q_0 \not\models^T \neg\varphi$.*

Proof For this proof, we reuse the game and formula of Fig. 3. Since the quantifier part is $\forall x.\ \exists y$, the timeline- and elementary semantics coincide for this formula. Since $\mathcal{G}, q_0 \models^E \varphi$, also $\mathcal{G}, q_0 \models^T \varphi$.

The negation of φ is

$$\neg\varphi = \exists x.\ \forall y.\ \bigvee \begin{cases} \neg\mathsf{assign}(\bigcirc \mapsto y).\ \mathbf{F}\, q_1 \\ \mathsf{assign}(\bigcirc \mapsto x).\ \mathbf{F}\, p_1 \wedge \neg\mathsf{assign}(\bigcirc \mapsto y).\ \mathbf{F}\, p_1 \\ \neg\mathsf{assign}(\bigcirc \mapsto x).\ \mathbf{F}\, p_1 \wedge \mathsf{assign}(\bigcirc \mapsto y).\ \mathbf{F}\, p_1. \end{cases}$$

Assume that there exists a timeline map $\overline{\theta}$ witnessing $\mathcal{G}, q_0 \models^T \neg\varphi$. Consider the valuations $w_1(y)(q_0) = w_2(y)(q_0) = q_2$, and $w_1(y)(q_0 \cdot q_2) = p_1$ and $w_2(y)(q_0 \cdot q_2) = p_2$. Notice that the first disjunct is not satisfied under those valuations. We consider two (symmetric) possiblities:

- we may have both $\theta(w_1)(x)(q_0)$ and $\theta(w_2)(x)(q_0)$ to q_1: then $\theta(w_1)(x)(q_0 \cdot q_1)$ and $\theta(w_2)(x)(q_0 \cdot q_1)$ must return the same move, since $w_1(y)(q_0) = w_2(y)(q_0)$. If they play to p_1, then none of the disjunct would be fulfilled under strategy valuation w_1; if they play to p_2, then all three disjunct are false under w_2.
- the argument is symmetric if $\theta(w_1)(x)(q_0) = \theta(w_2)(x)(q_0) = q_2$.

Hence $\mathcal{G}, q_0 \not\models^T \neg\varphi$ □

5 The Fragment SL[EG]$^\flat$

In this section, we focus on the timeline semantics $\models^T$. We exhibit a fragment[1] SL[EG]$^\flat$ of SL[BG]$^\flat$, containing SL[CG]$^\flat$ and SL[DG]$^\flat$, for which the syntactic and semantic negations coincide:

Theorem 1 *For any game $\mathcal{G}$ with initial state q_0, and any formula $\varphi \in$ SL[EG]$^\flat$, it holds $\mathcal{G}, q_0 \models^T \varphi \Leftrightarrow \mathcal{G}, q_0 \not\models^T \neg\varphi$.*

We prove this result in the sequel of this section. We first introduce *semi-stable* sets, which are the basis of the definition of SL[EG]$^\flat$; we then prove useful properties of those sets, and finally proceed to the proof of Theorem 1.

[1] We name our fragment SL[EG]$^\flat$ as it comes as a natural continuation after fragments SL[AG]$^\flat$ [33], SL[BG]$^\flat$ [30], and SL[CG]$^\flat$ and SL[DG]$^\flat$ [32].

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:

$$\overline{f} : i \mapsto 1 - f(i) \qquad f \curlywedge g \, : i \mapsto \min\{f(i), g(i)\} \qquad f \curlyvee g : i \mapsto \max\{f(i), g(i)\}.$$

The set $\{0,1\}^n$ can be seen as the lattice of subsets of $[1;n]$, with the above three operations corresponding to complement, intersection and union, respectively.

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

$$\forall s \in \{0,1\}^n. \qquad (f \curlywedge s) \curlyvee (g \curlywedge \overline{s}) \in F^n \text{ or } (g \curlywedge s) \curlyvee (f \curlywedge \overline{s}) \in F^n.$$

Example 3 Obviously, the set $\{0,1\}^n$ is semi-stable, as well as the empty set. It is easily seen that any singleton set also is semi-stable. 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 $(f \curlywedge s) \curlyvee (g \curlywedge \overline{s}) = (0,0)$ and $(g \curlywedge s) \curlyvee (f \curlywedge \overline{s}) = (1,1)$. Similarly, $\{(0,0),(1,1)\}$ is not semi-stable. Any other subset of $\{0,1\}^2$ is semi-stable.

We can now define SL[EG]$^\flat$ as follows:

$$\begin{aligned} \mathsf{SL[EG]}^\flat \ni \varphi &::= \forall x.\varphi \mid \exists x.\varphi \mid \xi & \xi &::= F^n((\omega_i)_{1 \le i \le n}) \\ \omega &::= \mathsf{assign}(\sigma).\ \psi & \psi &::= \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\,\psi \mid \psi\,\mathbf{U}\,\psi \mid p \end{aligned}$$

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

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

Equivalently:

$$\mathcal{G}, q \models_\chi F^n((\omega_i)_{i \le n}) \;\Leftrightarrow\; \mathcal{G}, q \models_\chi \bigvee_{f \in F^n} \left[\bigwedge_{f(i)=1} \omega_i \wedge \bigwedge_{f(i)=0} \neg\omega_i \right],$$

so that SL[EG]$^\flat$ is indeed a fragment of SL[BG]$^\flat$. Notice that SL[CG]$^\flat$ corresponds to the case where $F^n = \{\mathbf{1}^n\}$, which is semi-stable, so that SL[EG]$^\flat$ encompasses SL[CG]$^\flat$. As we prove later, $\{0,1\}^n \setminus \{\mathbf{0}^n\}$ also is semi-stable, which entails that SL[EG]$^\flat$ also subsumes SL[DG]$^\flat$.

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:

$$\exists x_1.\exists x_2.\forall y_1.\forall y_2.\ \bigwedge \begin{cases} (\mathsf{assign}(A_1 \mapsto y_1;\ A_2 \mapsto x_2).\psi_1) \Rightarrow (\mathsf{assign}(A_1 \mapsto x_1;\ A_2 \mapsto x_2).\psi_1) \\ (\mathsf{assign}(A_1 \mapsto x_1;\ A_2 \mapsto y_2).\psi_2) \Rightarrow (\mathsf{assign}(A_1 \mapsto x_1;\ A_2 \mapsto x_2).\psi_2) \end{cases}$$

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 $(f \curlywedge s) \curlyvee (g \curlywedge \overline{s}) = (1, 0, 0, 1)$ and $(g \curlywedge s) \curlyvee (f \curlywedge \overline{s}) = (0, 1, 1, 0)$, none of which is in F^4. Hence our formula is not (syntactically) in SL[EG]$^\flat$ (notice however that the existence of a Nash equilibrium can also be written as the disjunction (over all possible payoffs for the agents) of formulas in SL[CG]$^\flat$).

The definition of SL[EG] may look artificial. The main reason why we work with SL[EG] is that it is maximal for the first claim of Theorem 1 (see Prop. 9). But as the next result shows, it is actually a large fragment encompassing SL[AG] (hence also SL[CG] and SL[DG]):

Proposition 7 *SL[EG]$^\flat$ contains SL[AG]$^\flat$. The inclusion is strict (syntactically).*

Proof Remember that boolean combinations in SL[AG]$^\flat$ follow the grammar $\xi ::= \xi \vee \omega \mid \xi \wedge \omega \mid \omega$. In terms of subsets of $\{0, 1\}^n$, it corresponds to considering sets defined in one of the following two forms:

$$F^n_\xi = \{f \in \{0, 1\}^n \mid f(n) = 1\} \cup \{g \in \{0, 1\}^n \mid g_{|[1;n-1]} \in F^{n-1}_{\xi'}\}$$
$$F^n_\xi = \{f \in \{0, 1\}^n \mid f(n) = 1 \text{ and } f_{|[1;n-1]} \in F^{n-1}_{\xi'}\}$$

depending whether $\xi(p_j)_j = \xi'(p_j)_j \vee p_n$ or $\xi(p_j)_j = \xi'(p_j)_j \wedge p_n$. Assuming (by induction) that $F^{n-1}_{\xi'}$ is semi-stable, then we can prove that F^n_ξ also is. We detail the proof for the second case, the first case being similar.

Consider the case where $F^n_\xi = \{f \in \{0, 1\}^n \mid f(n) = 1 \text{ and} f_{|[1;n-1]} \in F^{n-1}_{\xi'}\}$. Pick any two elements f and g in F^n_ξ, and $s \in \{0, 1\}^n$. Since $f(n) = g(n) = 1$, we have $[(f \curlywedge s) \curlyvee (g \curlywedge \overline{s})](n) = [(f \curlywedge \overline{s}) \curlyvee (g \curlywedge s)](n) = 1$. Moreover, the restriction of $[(f \curlywedge s) \curlyvee (g \curlywedge \overline{s})]$ and of $[(f \curlywedge \overline{s}) \curlyvee (g \curlywedge s)]$ to their first $n - 1$ bits is computed from the restriction of f, g and s to their first $n - 1$ bits. Since $F^{n-1}_{\xi'}$ is semi-stable, one of $[(f \curlywedge s) \curlyvee (g \curlywedge \overline{s})]_{[1;n-1]}$ and $[(f \curlywedge \overline{s}) \curlyvee (g \curlywedge s)]_{[1;n-1]}$ belongs to $F^{n-1}_{\xi'}$, so that one of $[(f \curlywedge s) \curlyvee (g \curlywedge \overline{s})]$ and $[(f \curlywedge \overline{s}) \curlyvee (g \curlywedge s)]$ is in F^n_ξ.

That the inclusion is strict is proven by considering the semi-stable set $H^3 = \{(1, 1, 1), (1, 1, 0), (1, 0, 1),)(0, 1, 1)\}$. Assume that it corresponds to a formula in SL[AG]$^\flat$: then the boolean combination $\xi(x_1, x_2, x_3)$ of that formula must be in one of the following forms:

$$\xi'(x_1, x_2) \wedge x_3 \qquad \xi'(x_1, x_2) \vee x_3 \qquad \xi'(x_1, x_2) \wedge \neg x_3 \qquad \xi'(x_1, x_2) \vee \neg x_3.$$

It remains to prove that none of these cases corresponds to H^3: the first case does not allow $(1, 1, 0)$; the second case allows $(0, 0, 1)$; the third case does not allow $(1, 0, 1)$; the last case allows $(0, 0, 0)$. □

5.2 Properties of semi-stable Sets

Before proving our main theorem, we show that semi-stable sets enjoy several nice structural properties. Our first lemma entails that $\mathsf{SL[EG]}^\flat$ is closed under (syntactic) negation.

Lemma 3 *F^n is semi-stable if, and only if, its complement is.*

Proof Assume F^n is not semi-stable, and pick f and g in F^n and $s \in \{0,1\}^n$ such that none of $\alpha = (f \curlywedge s) \curlyvee (g \curlywedge \overline{s})$ and $\gamma = (g \curlywedge s) \curlyvee (f \curlywedge \overline{s})$ are in F^n. It cannot be the case that $g = f$, as this would imply $\alpha = f \in F^n$. Hence $\alpha \neq \gamma$. We claim that α and γ are our witnesses for showing that the complement of F^n is not semi-stable: both of them belong to the complement of F^n, and $(\alpha \curlywedge s) \curlyvee (\gamma \curlywedge \overline{s})$ can be seen to equal f, hence it is not in the complement of F^n. Similarly for $(\gamma \curlywedge s) \curlyvee (\alpha \curlywedge \overline{s}) = g$. □

Lemma 4 *If $F^n \subseteq \{0,1\}^n$ is semi-stable, then for any $s \in \{0,1\}^n$ and any non-empty subset H^n of F^n, it holds that*

$$\exists f \in H^n.\, \forall g \in H^n.\, (f \curlywedge s) \curlyvee (g \curlywedge \overline{s}) \in F^n.$$

Proof For a contradiction, assume that there exist $s \in \{0,1\}^n$ and $H^n \subseteq F^n$ such that, for any $f \in H^n$, there is an element $g \in H^n$ for which $(f \curlywedge s) \curlyvee (g \curlywedge \overline{s}) \notin F^n$. Then there must exist a minimal integer $2 \leq \lambda \leq |H^n|$ and λ elements $\{f_i \mid 1 \leq i \leq \lambda\}$ of H^n such that

$$\forall 1 \leq i \leq \lambda - 1\ (f_i \curlywedge s) \curlyvee (f_{i+1} \curlywedge \overline{s}) \notin F^n \text{ and } (f_\lambda \curlywedge s) \curlyvee (f_1 \curlywedge \overline{s}) \notin F^n.$$

By Lemma 3, the complement of F^n is semi-stable. Hence, considering $(f_{\lambda-1} \curlywedge s) \curlyvee (f_\lambda \curlywedge \overline{s})$ and $(f_\lambda \curlywedge s) \curlyvee (f_1 \curlywedge \overline{s})$, one of the following two vectors is not in F^n:

$$([(f_{\lambda-1} \curlywedge s) \curlyvee (f_\lambda \curlywedge \overline{s})] \curlywedge s) \curlyvee ([(f_\lambda \curlywedge s) \curlyvee (f_1 \curlywedge \overline{s})] \curlywedge \overline{s})$$
$$([(f_\lambda \curlywedge s) \curlyvee (f_1 \curlywedge \overline{s})] \curlywedge s) \curlyvee ([(f_{\lambda-1} \curlywedge s) \curlyvee (f_\lambda \curlywedge \overline{s})] \curlywedge \overline{s})$$

The second expression equals f_λ, which is in F^n. Hence we get that $(f_{\lambda-1} \curlywedge s) \curlyvee (f_1 \curlywedge \overline{s})$ is not in F^n, contradicting minimality of λ. □

For two elements f and g of $\{0,1\}^n$, we write $f \leq g$ whenever $f(i) = 1$ implies $g(i) = 1$ for all $i \in [1,n]$ (this corresponds to set inclusion when seeing $\{0,1\}^n$ as the lattice of subsets of $[1;n]$). Given $B^n \subseteq \{0,1\}^n$, we write $\uparrow B^n = \{g \in \{0,1\}^n \mid \exists f \in B^n,\ f \leq g\}$. A set $F^n \subseteq \{0,1\}^n$ is *upward-closed* if $F^n = \uparrow F^n$. Notice that being upward-closed and being semi-stable are uncomparable (for instance, the set $\uparrow\{(0,0,1,1); (1,1,0,0)\}$ is not semi-stable). We now explain how to transform a semi-stable set into an upward-closed one by flipping some of its bits. This will simplify the presentation of the proof of our main theorem.

Fix a vector $b \in \{0,1\}^n$. We define the operation $\mathsf{flip}_b\colon \{0,1\}^n \to \{0,1\}^n$ that maps any vector f to $(f \curlywedge b) \curlyvee (\overline{f} \curlywedge \overline{b})$. In other terms, flip_b flips the i-th bit of its argument if $b_i = 0$, and keeps this bit unchanged if $b_i = 1$. In $\mathsf{SL[EG]}^\flat$, flipping bits amounts to negating the corresponding goals. The first part of the following lemma thus indicates that our definition for $\mathsf{SL}[\mathsf{EG}]^\flat$ is sound.

Lemma 5 *For any $b \in \{0,1\}^n$, if $F^n \subseteq \{0,1\}^n$ is semi-stable, then so is $\mathsf{flip}_b(F^n)$. Moreover, for any semi-stable set F^n, there exists $b \in \{0,1\}^n$ such that $\mathsf{flip}_b(F^n)$ is upward-closed.*

Example 5 Take $F^2 = \{(0,0),(1,0),(1,1)\}$. This set is semi-stable, but it is not upward-closed. Letting $b = (1,0)$, we have $\mathsf{flip}_b(F^2) = \{(0,1),(1,1),(1,0)\}$, which is upward-closed (and still semi-stable).

Proof We begin with the first statement. Assume that F^n is semi-stable, and take $f' = \mathsf{flip}_b(f)$ and $g' = \mathsf{flip}_b(g)$ in $\mathsf{flip}_b(F^n)$, and $s \in \{0,1\}^n$. By distributivity, we get

$$\begin{aligned}(f' \curlywedge s) \curlyvee (g' \curlywedge \overline{s}) &= \big(((f \curlywedge b) \curlyvee (\overline{f} \curlywedge \overline{b})) \curlywedge s\big) \curlyvee \big(((g \curlywedge b) \curlyvee (\overline{g} \curlywedge \overline{b})) \curlywedge \overline{s}\big) \\ &= (((f \curlywedge s) \curlyvee (g \curlywedge \overline{s})) \curlywedge b) \curlyvee \big(((\overline{f} \curlywedge s) \curlyvee (\overline{g} \curlywedge \overline{s})) \curlywedge \overline{b}\big)\end{aligned}$$

Write $\alpha = (f \curlywedge s) \curlyvee (g \curlywedge \overline{s})$ and $\beta = (\overline{f} \curlywedge s) \curlyvee (\overline{g} \curlywedge \overline{s})$. One can easily check that $\beta = \overline{\alpha}$. We then have

$$(f' \curlywedge s) \curlyvee (g' \curlywedge \overline{s}) = (\alpha \curlywedge b) \curlyvee \big(\overline{\alpha} \curlywedge \overline{b}\big) = \mathsf{flip}_b(\alpha). \tag{4}$$

This computation being valid for any f and g, we also have

$$(g' \curlywedge s) \curlyvee (f' \curlywedge \overline{s}) = (\gamma \curlywedge b) \curlyvee \big(\overline{\gamma} \curlywedge \overline{b}\big) = \mathsf{flip}_b(\gamma) \tag{5}$$

with $\gamma = (g \curlywedge s) \curlyvee (f \curlywedge \overline{s})$. By hypothesis, at least one of α and γ belongs to F^n, so that also at least one of $(f' \curlywedge s) \curlyvee (g' \curlywedge \overline{s})$ and $(g' \curlywedge s) \curlyvee (f' \curlywedge \overline{s})$ belongs to $\mathsf{flip}_b(F^n)$.

The second statement of Lemma 5 trivially holds for $F^n = \emptyset$; thus in the following, we assume F^n to be non-empty. For $1 \leq i \leq n$, let $s_i \in \{0,1\}^n$ be the vector such that $s_i(j) = 1$ if, and only if, $j = i$. Applying Lemma 4, we get that for any i, there exists some $f_i \in F^n$ such that for any $f \in F^n$, it holds

$$(f_i \curlywedge s_i) \curlyvee (f \curlywedge \overline{s}_i) \in F^n. \tag{6}$$

We fix such a family $(f_i)_{i \leq n}$ then define $g \in \{0,1\}^n$ as $g = \bigcurlyvee_{1<i<n}(f_i \curlywedge s_i)$, i.e. $g(i) = f_i(i)$ for all $1 \leq i \leq n$. Starting from any element of F^n and applying (6) iteratively for each i, we get that $g \in F^n$. Since $g \curlywedge s_i = f_i \curlywedge s_i$, we also have

$$\forall f \in F^n \qquad (g \curlywedge s_i) \curlyvee (f \curlywedge \overline{s}_i) \in F^n$$

By (5), since $\mathsf{flip}_g(g) = \mathbf{1}$, we get

$$\forall f \in F^n \qquad (\mathbf{1} \curlywedge s_i) \curlyvee (\mathsf{flip}_g(f) \curlywedge \overline{s}_i) \in \mathsf{flip}_g(F^n). \tag{7}$$

Now, assume that $\mathsf{flip}_g(F^n)$ is not upward closed: then there exist elements $f \in F^n$ and $h \notin F^n$ such that $\mathsf{flip}_g(f)(i) = 1 \Rightarrow \mathsf{flip}_g(h)(i) = 1$ for all i. Starting from f and iteratively applying (7) for those i for which $\mathsf{flip}_g(h)(i) = 1$ and $\mathsf{flip}_g(f)(i) = 0$, we get that $\mathsf{flip}_g(h) \in \mathsf{flip}_g(F^n)$ and $h \in F^n$. Hence $\mathsf{flip}_g(F^n)$ must be upward closed. □

5.3 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

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

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 6 *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 $\mathbb{F}^n(h_2,s) \subseteq \mathbb{F}^n(h_1,s)$.*

Proof Assuming otherwise, there would exist $h'_1 \in \mathbb{F}^n(h_1,s)\backslash\mathbb{F}^n(h_2,s)$ and $h'_2 \in \mathbb{F}^n(h_2,s)\backslash\mathbb{F}^n(h_1,s)$. We then have:

$$\begin{array}{ll}(h_1 \curlywedge s) \curlyvee (h'_1 \curlywedge \overline{s}) \in F^n & (h_2 \curlywedge s) \curlyvee (h'_1 \curlywedge \overline{s}) \notin F^n\\ (h_2 \curlywedge s) \curlyvee (h'_2 \curlywedge \overline{s}) \in F^n & (h_1 \curlywedge s) \curlyvee (h_2 \curlywedge \overline{s}) \notin F^n\end{array}$$

Now consider $(h_1 \curlywedge s) \curlyvee (h'_1 \curlywedge \overline{s})$, $(h_2 \curlywedge s) \curlyvee (h'_2 \curlywedge \overline{s})$ and s. As F^n is semi-stable, one of the two following vector is in F^n :

$$\begin{array}{l}\big((h_1 \curlywedge s) \curlyvee (h'_1 \curlywedge \overline{s}) \curlywedge s\big) \curlyvee \big((h_2 \curlywedge s) \curlyvee (h'_2 \curlywedge \overline{s}) \curlywedge \overline{s}\big)\\ \big((h_2 \curlywedge s) \curlyvee (h'_2 \curlywedge \overline{s}) \curlywedge s\big) \curlyvee \big((h_1 \curlywedge s) \curlyvee (h'_1 \curlywedge \overline{s}) \curlywedge \overline{s}\big)\end{array}$$

The first vector is equal to $(h_1 \curlywedge s) \curlyvee (h'_2 \curlywedge \overline{s})$ and the second to $(h_2 \curlywedge s) \curlyvee (h'_1 \curlywedge \overline{s})$ and both are supposed to be in $\overline{F^n}$, we get a contradiction. □

Given a semi-stable set F^n and $s \in \{0,1\}^n$, we can use the inclusion relation of Lemma 6 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 reflexiveness and transitivity both follow from the reflexiveness and transitivity of the inclusion relation $\subseteq$. By Lemma 6, 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 $(h_1 \curlywedge s) = (h_2 \curlywedge s)$, we have $\mathbb{F}(h_1,s) = \mathbb{F}(h_2,s)$, and $h_1 \equiv_s h_2$.

Example 6 Consider the set $F^3 = \{(1,0,0),(1,1,0),(1,0,1),(0,1,1),(1,1,1)\}$ represented on Fig. 7, which can be shown to be semi-stable.

Fix $s = (1,1,0)$. Then $\mathbb{F}^3((0,1,\star),s) = \{0,1\}^2 \times \{1\}$: the only way to complete $(0,1,\star)$ to an element in F^3 is by replacing $\star$ with 1. Similarly, $\mathbb{F}^3((1,1,\star),s) = \mathbb{F}^3((1,0,\star),s) = \{0,1\}^3$, and $\mathbb{F}^3((0,0,\star),s) = \varnothing$. It follows that $(0,0,\star) \preceq_s (0,1,\star) \preceq_s (1,0,\star) \equiv_s (1,1,\star)$.

For $s' = (0,0,1)$, we can proceed similarly and get that $(\star,\star,0) \preceq_{s'} (\star,\star,1)$.

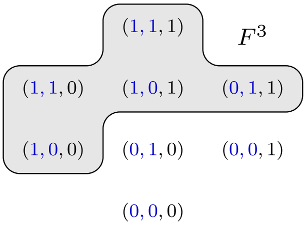


Fig. 7 A semi-stable set over $\{0, 1\}^n$

We now prove a technical result over such orders, which will be useful for the proof of Lemma 11.

Lemma 7 *Given a semi-stable set F^n, $s_1, s_2 \in \{0, 1\}^n$ such that $s_1 \curlywedge s_2 = \boldsymbol{0}$ and $f, g \in \{0, 1\}^n$ such that $f \preceq_{s_1} g$ and $f \preceq_{s_2} g$, it holds $f \preceq_{s_1 \curlyvee s_2} g$.*

Example 7 Consider again the semi-stable set F^3 of Example 6. Observe that for $s_1 = (1, 0, 0)$, it holds $(0, \star, \star) \preceq_{s_1} (1, \star, \star)$, because for any $x, y \in \{0, 1\}$, if $(0, x, y) \in F^3$, then also $(1, x, y) \in F^3$; similarly, for $s_2 = (0, 1, 0)$, we have $(\star, 0, \star) \preceq_{s_2} (\star, 1, \star)$. Lemma 7 entails that $(0, 0, \star) \preceq_s (1, 1, \star)$, with $s = (1, 1, 0)$.

Proof Because $f \preceq_{s_1} g$ and $f \preceq_{s_2} g$, we have

$$\forall i \in \{1, 2\} \forall h \in \{0, 1\}^n \; (f \curlywedge s_i) \curlyvee (h \curlywedge \overline{s_i}) \in F^n \Rightarrow (g \curlywedge s_i) \curlyvee (h \curlywedge \overline{s_i}) \in F^n \quad (8)$$

Consider $h' \in \{0, 1\}^n$ such that $\alpha = (f \curlywedge (s_1 \curlyvee s_2)) \curlyvee (h' \curlywedge \overline{(s_1 \curlyvee s_2)})$ is in F^n. Define the element $h = \alpha \curlywedge \overline{s_2}$, then $(f \curlywedge s_2) \curlyvee (h \curlywedge \overline{s_2}) = (f \curlywedge (s_1 \curlyvee s_2)) \curlyvee (h' \curlywedge \overline{(s_1 \curlyvee s_2)}) \in F^n$. Using (8) with s_2 and h, we get $\beta = (g \curlywedge s_2) \curlyvee (h \curlywedge \overline{s_2})$. As $s_1 \curlywedge s_2 = \mathbf{0}$, we can write $\beta = (f \curlywedge s_1) \curlyvee (g \curlywedge s_2) \curlyvee (h' \curlywedge \overline{(s_1 \curlyvee s_2)}) \in F^n$.

Now consider $h = \beta \curlywedge \overline{s_1}$, we have $(f \curlywedge s_1) \curlyvee (h \curlywedge \overline{s_1}) = \beta \in F^n$. Using (8) with s_1 and h, we get $(g \curlywedge (s_1 \curlyvee s_2)) \curlyvee (h' \curlywedge \overline{(s_1 \curlyvee s_2)}) \in F^n$. Therefore $\mathbb{F}^n(f, s_1 \curlyvee s_2) \subseteq \mathbb{F}^n(g, s_1 \curlyvee s_2)$ and $f \preceq_{s_1 \curlyvee s_2} g$. □

The following lemma is straightforward:

Lemma 8 *Assuming F^n is upward-closed, for any f, g and s in $\{0, 1\}^n$, if $f \leq g$ (i.e. for all i, $f(i) = 1 \Rightarrow g(i) = 1$), then $f \preceq_s g$. In particular, $\boldsymbol{0}$ is a minimal element for $\preceq_s$, for any s.*

Proof Since $f \leq g$, then also $(f \curlywedge s) \curlyvee (h \curlywedge \overline{s}) \leq (g \curlywedge s) \curlyvee (h \curlywedge \overline{s})$, for any $h \in \{0, 1\}^n$. Since F^n is upward-closed, if $(f \curlywedge s) \curlyvee (h \curlywedge \overline{s})$ is in F^n, then so is $(g \curlywedge s) \curlyvee (h \curlywedge \overline{s})$. □

5.4 Sketch of Proof of Theorem 1

The proof of Theorem 1 is long and technical. Before giving the full details, we begin with some intuition how semi-stable sets, and the quasi-orders defined above, are used to prove the result. We first notice that the approach we used in Prop. 2 does not extend in general to formulas with several goals. Consider for instance formula $(Q_i x_i)_{i \le l}(\beta_1. \varphi_1 \Leftrightarrow \beta_2. \varphi_2)$: if at some points the two goals give rise to two different outcomes, thus to two different subgames, the winning objectives in one subgame depends on what is achieved in the other subgame.

$\mathsf{SL[EG]}^\flat$ has been designed to simplify such dependences between different subgames: 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 $\mathsf{SL[EG]}^\flat$ formula being checked. Consider again Example 6: associating the set F^3 with three goals ω_1, ω_2 and ω_3 (and adequate strategy quantifiers), we get a formula in $\mathsf{SL[EG]}^\flat$. Assume that the moves selected by the players give rise to the same transition for ω_1 and ω_2, and to a different transition for ω_3; this gives rise to two subgames. In the subgame reached when following the transition of ω_1 and ω_2 (hence with $s = (1, 1, 0)$), the optimal way of playing 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 ω_3; for instance, it is better to fulfill only ω_1 than to fulfill only ω_2 (i.e. $(0, 1, \star) \preceq_s (1, 0, \star)$), which can be observed on Fig. 7 by the fact that fulfilling ω_1 is enough to make the whole formula hold true. In the subgame corresponding to ω_3, the optimal way of playing is given by $(\star, \star, 0) \preceq_{s'} (\star, \star, 1)$: it is always better to fulfill ω_3, whatever happens on the other subgame.

Our proof follows the schema depicted on Fig. 8. Building on the idea depicted on Fig. 4, we would like to construct a turn-based parity game encoding the $\mathsf{SL[EG]}^\flat$ model-checking instance at hand. Strategy quantifiers are encoded with tree-shaped *quantification games* as in Fig. 4, but now, the leaves of quantification games may give rise to different outcomes, depending on the goal being considered: Fig. 8 depicts the case of a leaf from which the first two goals would go in one direction (to q_1 here) while the third goal follows a different direction (to q_2). Notice that from the other leaves, the goals may have been grouped differently (and in particular, they may have all given rise to the same transition).

Now, consider the outcome generated by the first two goals: it goes to a subgame starting in state q_1, and only the first two goals have to be tracked. From our observations above, we can compute an order defining the best way of satisfying the remaining two goals; this does not depend on what happens along the other outcome, generated by the third goal. We can thus consider this subgame alone, and apply the same construction with the remaining goals (using parity automata to keep track of the satisfaction of the LTL formulas in the goals). Since there are finitely many goals, we eventually end up in a situation where there is a single goal, or where the goals always give rise to the same outcomes; then the computation remains in the same subgame, and the situation corresponds to the case of Fig. 4.

We implement these ideas as follows: 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* in that subgame. For each subgame, we can compute, by induction on s, the optimal set

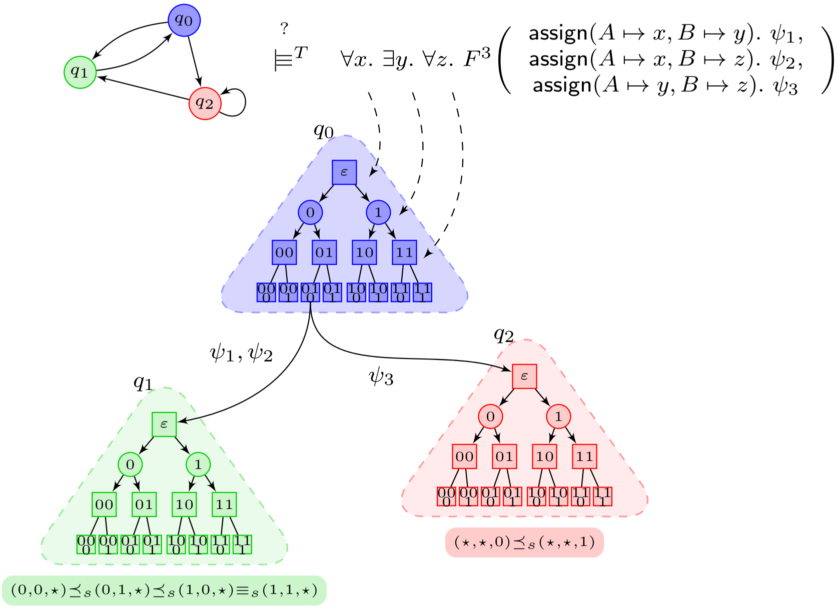

q_0
q_1
q_2
$\overset{?}{\models}^T \quad \forall x.\ \exists y.\ \forall z.\ F^3\left(\begin{array}{l}\mathsf{assign}(A \mapsto x, B \mapsto y).\ \psi_1,\\ \mathsf{assign}(A \mapsto x, B \mapsto z).\ \psi_2,\\ \mathsf{assign}(A \mapsto y, B \mapsto z).\ \psi_3\end{array}\right)$
ε
0
1
00
01
10
11
ψ_1, ψ_2
ψ_3
$(0,0,\star)\preceq_S(0,1,\star)\preceq_S(1,0,\star)\equiv_S(1,1,\star)$
$(\star,\star,0)\preceq_S(\star,\star,1)$

Fig. 8 In a formula based on the semi-stable sets of Fig. 7, upon separation of the goals, the game splits into independent subgames

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 $\overline{\theta}$; using similar arguments as for the proof of Prop. 5, we get a valuation χ such that $\theta(\chi_{|\mathcal{V}^\forall}) = \chi = \overline{\theta}(\chi_{|\mathcal{V}^\exists})$, from which we deduce that exactly one of φ and $\neg\varphi$ holds.

5.5 Proof of Theorem 1

We can now prove our main theorem, which we first restate:

Theorem 1 *For any game $\mathcal{G}$ with initial state q_0, and any formula $\varphi \in$, SL[EG]$^\flat$ it holds $\mathcal{G}, q_0 \models^T \varphi \Leftrightarrow \mathcal{G}, q_0 \not\models^T \neg\varphi$.*

Proof Following Lemma 5, we assume for the rest of the proof that the set F^n of the SL[EG]$^\flat$ formula φ is upward-closed (even if it means negating some of the LTL objectives). We also assume it is non-empty, since the result is trivial otherwise.

The proof of Theorem 1 is in three steps:

- we build a family of parity automata expressing the objectives that may have to be fulfilled along outcomes. A configuration of a subgame is then described by a state q of the game, a vector d of states of those parity automata, and a set s of goals that are still *active* in the current subgame;

- we characterize the two ways of fulfilling a set of goals: either by fulfilling all goals along the same outcome, or by partitioning them among different branches;
- we encode these two possibilities into 2-player parity games, and inductively compute optimal sets of goals (represented as vectors $b_{q,d,s} \in \{0,1\}^n$) that can be achieved from any given configuration. By determinacy of parity games, we derive timeline maps witnessing the fact that $b_{q,d,s}$ can be achieved, and the fact that it is optimal. If $b_{q_0,d_0,\mathbf{1}} \in F^n$, we get a witness map for $\mathcal{G}, q_0 \models^T \varphi$; otherwise, we get one for $\mathcal{G}, q_0 \models^T \neg\varphi$. □

5.5.1 Automata for Conjunctions of Goals

We use *deterministic parity word automata* to keep track of the goals to be satisfied. Since we initially have no clue about which goal(s) will have to be fulfilled along an outcome, we use a (large) set of automata, all running in parallel.

For $s \in \{0,1\}^n$ and $h \in \{0,1\}^n$, we let $D_{s,h}$ be a deterministic parity automaton accepting exactly the words over 2^{AP} along which the following formula $\Phi_{s,h}$ holds:

$$\Phi_{s,h} = \bigvee_{\substack{k\in\{0,1\}^n \\ h\preceq_s k}} \bigwedge_{\substack{j \text{ s.t.} \\ (k\curlywedge s)(j)=1}} \varphi_j.$$

where a conjunction over an empty set (i.e., if $(k \curlywedge s)(j) = 0$ for all j) is true.

Notice that in $\Phi_{s,h}$, we should also have imposed $\neg\varphi_j$ for those indices j for which $(k \curlywedge s)(j) = 0$. However, using Lemma 8, if $h \preceq_s k$ and $k \leq k'$, then also $h \preceq_s k'$, so that any conjunction containing more φ_j's would also appear in $\Phi_{s,h}$.

Notice that when $s = \mathbf{0}$, we have $h \preceq_s k$ for any h and k, so that $\Phi_{\mathbf{0},h}$ is true for any $h \in \{0,1\}^n$). From now on, we only consider vectors $s \in \{0,1\}^n$ such that $|s| = \sum_{1\leq i\leq n} s_i \geq 1$.

As an example, take $s \in \{0,1\}^n$ with $|s = 1|$, writing j for the index where $s(j) = 1$; for any $h \in \{0,1\}^n$, if there is $k \succeq_s h$ with $k(j) = 0$ (which in particular is the case when $h(j) = 0$), then the automaton $D_{s,h}$ is universal; otherwise $D_{s,h}$ accepts the set of words over 2^{AP} along which φ_j holds.

We write $\mathcal{D} = \{D_{s,h} \mid s \in \{0,1\}^n,\ h \in \{0,1\}^n\}$ for the set of automata defined above. A *vector of states of* $\mathcal{D}$ is a function associating with each automaton $D \in \mathcal{D}$ one of its states. We write VS for the set of all vectors of states of $\mathcal{D}$. For any vector $d \in \mathsf{VS}$ and any state q of $\mathcal{G}$, we let $\mathsf{succ}(d,q)$ to be the vector of states associating with each $D \in \mathcal{D}$ the successor of state $d(D)$ after reading $\mathsf{lab}(q)$; we extend succ to finite paths $(q_i)_{0\leq i\leq n}$ in $\mathcal{G}$ inductively, letting $\mathsf{succ}(d,(q_i)_{0\leq i\leq n}) = \mathsf{succ}(\mathsf{succ}(d,(q_i)_{0\leq i\leq n-1}), q_n)$.

An infinite path $(q_i)_{i\in\mathbb{N}}$ in $\mathcal{G}$ is accepted by an automaton D of $\mathcal{D}$ whenever the word $(\mathsf{lab}(q_i))_{i\in\mathbb{N}}$ is accepted by D. We write $\mathcal{L}(D)$ for the set of paths of $\mathcal{G}$ accepted by D. Finally, for $d \in \mathsf{VS}$, we write $\mathcal{L}(D^d_{s,h})$ for the set of words that are accepted by $D_{s,h}$ starting from the state $d(D_{s,h})$ of $D_{s,h}$.

Proposition 8 *The following holds for any $s \in \{0,1\}^n$:*

1. $\Phi_{s,\mathbf{0}} \equiv \top$ *(i.e., $D_{s,\mathbf{0}}$ is universal);*

2. *for any* $h_1, h_2 \in \{0,1\}^n$, *if* $h_1 \preceq_s h_2$, *we have* $\Phi_{s,h_2} \Rightarrow \Phi_{s,h_1}$ *(i.e.,* $\mathcal{L}(D_{s,h_2}) \subseteq \mathcal{L}(D_{s,h_1})$*);*
3. *for any* $h \in F^n$, $\Phi_{\boldsymbol{1},h} \equiv \bigvee_{k \in F^n} \bigwedge_{j \text{ s.t. } k(j)=1} \varphi_j$.

Proof $\Phi_{s,\mathbf{0}}$ contains the empty conjunction ($k = \mathbf{0}$) as a disjunct. Hence it is equivalent to true. When $h_1 \preceq_s h_2$, formula Φ_{s,h_1} contains more disjuncts than Φ_{s,h_2}, hence the second result. Finally, $\mathbb{F}^n(f, \mathbf{1}) = \{0,1\}^n$ if $f \in F^n$, and is empty otherwise. Hence if $h \in F^n$, we have $h \preceq_{\mathbf{1}} k$ if, and only if, $k \in F^n$, which entails the result. □

5.5.2 Two Ways of Achieving Goals

After a given history, a set of goals may be achieved either along a single outcome, in case the assignment of strategies to players gives rise to the same outcomes, or they may be split among different outcomes. We express those two ways of satisfying goals, by means of two operators parameterized by the current configuration.

The first operator covers the case where the goals currently enabled by s (those goals $\beta_i.\ \varphi_i$ for which $s(i) = 1$) are all fulfilled along the same outcome. For any $d \in \mathsf{VS}$ and any two s and h in $\{0,1\}^n$, the operator $\Gamma^{\mathsf{stick}}_{d,s,h}$ is defined as follows: given a context χ with $\mathcal{V} \subseteq \mathsf{dom}(\chi)$ and a state q of $\mathcal{G}$,

$$\mathcal{G}, q \models_\chi \Gamma^{\mathsf{stick}}_{d,s,h} \Leftrightarrow \exists \rho \in \mathsf{Play}_{\mathcal{G}}(q) \text{ s.t.} \begin{cases} -\forall j \le n.\ \big(s(j) = 1 \Rightarrow \mathsf{out}(q, \chi \circ \beta_j) = \rho\big) \\ -\rho \in \mathcal{L}(D^d_{s,h}) \end{cases}$$

Intuitively, all the goals enabled by s must give rise to the same outcome, which is accepted by $D^d_{s,h}$. In the formula above, $\chi \circ \beta_j$ corresponds to the strategy profile to be used for goal $\beta_j \cdot \varphi_j$.

We now consider the case where the active goals are partitioned among different outcomes.

Definition 2 A *partition* of an element $s \in \{0,1\}^n$ is a sequence $(s_\kappa)_{1 \le \kappa \le \lambda}$, with $\lambda \ge 2$, of elements of $\{0,1\}^n$ with $s_1 \curlyvee \cdots \curlyvee s_\lambda = s$ and where for any two $\kappa \ne \kappa'$, $s_\kappa \curlywedge s_{\kappa'} = \mathbf{0}$.

An *extended* partition of s is a sequence $\tau = (s_\kappa, q_\kappa, d_\kappa)_{1 \le \kappa \le \lambda}$ of elements of $\{0,1\}^n \times \mathsf{Q} \times \mathsf{VS}$ where $(s_\kappa)_{1 \le \kappa \le \lambda}$ is a partition of s, q_κ are states of $\mathcal{G}$, and d_κ are vectors of states of the automata in $\mathcal{D}$.

We write $\mathsf{Part}(s)$ for the set of all extended partitions of s. Notice that we only consider non-trivial partitions; in particular, if $|s| \le 1$, then $\mathsf{Part}(s) = \emptyset$. For any $d \in \mathsf{VS}$, any s in $\{0,1\}^n$ and any set of partitions Υ_s of s, the operator $\Gamma^{\mathsf{sep}}_{d,s,\Upsilon_s}$ states that the goals currently enabled by s all follow a common history ρ for a finite number of steps, and then partition themselves according to some partition in Υ_s. The operator $\Gamma^{\mathsf{sep}}_{d,s,\Upsilon_s}$ is defined as follows:

$$\mathcal{G}, q \models_\chi \Gamma^{\mathsf{sep}}_{d,s,\Upsilon_s} \Leftrightarrow \begin{array}{l} \exists \tau \in \Upsilon_s. \\ \exists \rho \in \mathsf{Hist}_{\mathcal{G}}(q). \end{array} \begin{cases} -\forall j \le n.\ \big(s(j) = 1 \Rightarrow \rho \in \mathsf{Pref}(\mathsf{out}(q, \chi \circ \beta_j))\big) \\ -\forall \kappa \le |\tau|.\ \forall j \le n.\ \text{ letting } m_j(A) = \chi(\beta_j(A))(\rho). \\ \quad \big(s_\kappa(j) = 1 \Rightarrow q_\kappa = \Delta(\mathsf{last}(\rho), m_j)\big) \\ -\forall \kappa \le |\tau|.\ \mathsf{succ}(d, \rho \cdot q_\kappa) = d_\kappa. \end{cases}$$

Notice that h does not appear explicitly in this definition, but $\Gamma^{\mathsf{sep}}_{d,s,\Upsilon_s}$ will depend on h through the choice of Υ_s. The operators Γ^{stick} and Γ^{sep} are illustrated on Fig. 9.

5.5.3 Fulfilling Optimal Sets of Goals

We now inductively (on $|s|$) define new operators $\Gamma_{d,s,h}$ combining the above two operators Γ^{stick} and Γ^{sep}, and selecting optimal ways of partitioning the goals among the outcomes.

Base case: $|s| = 1$ When only one goal is enabled, we only have to consider a single outcome, so that we let $\Gamma_{d,s,h} = \Gamma^{\mathsf{stick}}_{d,s,h}$, for any $d \in \mathsf{VS}$ and $h \in \{0,1\}^n$. By Prop. 8, for any context χ such that $\mathsf{Agt} \subseteq \mathsf{dom}(\chi)$, it holds $\mathcal{G}, q \models_\chi \Gamma_{d,s,\mathbf{0}}$, hence also $\mathcal{G}, q \models^T (Q_i x_i)_{1\leq i\leq l}.\ \Gamma_{d,s,\mathbf{0}}$. Hence there must exist a maximal value b in the lattice $\{0,1\}^n$ such that $\mathcal{G}, q \models^T (Q_i x_i)_{1\leq i\leq l}.\ \Gamma_{d,s,b}$. We write $b_{q,d,s}$ for one such value (notice that it need not be unique). By maximality, for any h such that $b_{q,d,s} \prec_s h$, we have $\mathcal{G}, q \not\models^T (Q_i x_i)_{1\leq i\leq l}.\ \Gamma_{d,s,h}$.

Induction step We assume that for any $d \in \mathsf{VS}$, any $h \in \{0,1\}^n$ and any $s \in \{0,1\}^n$ with $|s \leq k|$, we have defined an operator $\Gamma_{d,s,h}$, and that for any $q \in \mathsf{Q}$, we have fixed an element $b_{q,d,s} \in \{0,1\}^n$ for which $\mathcal{G}, q \models^T (Q_i x_i)_{1\leq i\leq l}.\ \Gamma_{d,s,b}$ and such that for any h such that $b_{q,d,s} \prec_s h$, it holds $\mathcal{G}, q \not\models^T (Q_i x_i)_{1\leq i\leq l}.\ \Gamma_{d,s,h}$.

Pick $s \in \{0,1\}^n$ with $|s = k+1|$, together with an extended partition $\tau = (s_\kappa, q_\kappa, d_\kappa)_{1\leq\kappa\leq\lambda}$. Then we must have $|s_\kappa| < k+1$ for all $1 \leq \kappa \leq \lambda$, so that $\Gamma_{d_\kappa,s_\kappa,h}$ and $b_{q_\kappa,d_\kappa,s_\kappa}$ have been defined at previous steps. We let

$$c_{s,\tau} = \bigcurlyvee_{1\leq\kappa\leq\lambda} (s_\kappa \curlywedge b_{q_\kappa,d_\kappa,s_\kappa}).$$

We then define

$$\Gamma_{d,s,h} = \Gamma^{\mathsf{stick}}_{d,s,h} \vee \Gamma^{\mathsf{sep}}_{d,s,\Upsilon_{s,h}} \qquad \text{with } \Upsilon_{s,h} = \{\tau \in \mathsf{Part}(s) \mid h \preceq_s c_{s,\tau}\}.$$

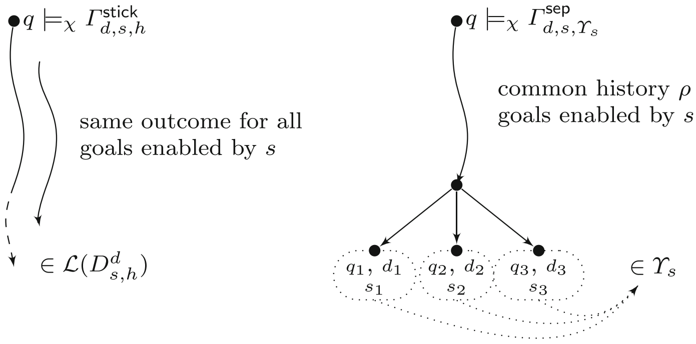

$q \models_\chi \Gamma^{\mathsf{stick}}_{d,s,h}$
same outcome for all goals enabled by s
$\in \mathcal{L}(D^d_{s,h})$
$q \models_\chi \Gamma^{\mathsf{sep}}_{d,s,\Upsilon_s}$
common history ρ for all goals enabled by s
q_1, d_1 s_1
q_2, d_2 s_2
q_3, d_3 s_3
$\in \Upsilon_s$

Fig. 9 Illustration of $\Gamma^{\mathsf{stick}}_{d,s,h}$ and $\Gamma^{\mathsf{sep}}_{d,s,\Upsilon_s}$

As previously, we claim that $\mathcal{G}, q \models_\chi \Gamma_{d,s,\mathbf{0}}$ for any χ such that $\mathsf{Agt} \subseteq \mathsf{dom}(\chi)$. Indeed, for a given χ, if all the outcomes of the goals enabled by s follow the same infinite path, then this path is accepted by $D_{s,\mathbf{0}}$ and $\mathcal{G}, q \models_\chi \Gamma^{\mathsf{stick}}_{d,s,\mathbf{0}}$; otherwise, after some common history ρ, the outcomes are partitioned following some extended partition τ_0, which obviously satisfies $\mathbf{0} \preceq_s c_{s,\tau_0}$ since $\mathbf{0}$ is a minimal element of $\preceq_s$. Hence in that case $\mathcal{G}, q \models_\chi \Gamma^{\mathsf{sep}}_{d,s,\Upsilon_{s,\mathbf{0}}}$.

In particular, it follows that $\mathcal{G}, q \models^T (Q_i x_i)_{1\le i\le l}.\ \Gamma_{d,s,\mathbf{0}}$, and we can fix a maximal element $b_{q,d,s}$ for which $\mathcal{G}, q \models^T (Q_i x_i)_{1\le i\le l}.\ \Gamma_{d,s,b_{q,d,s}}$ and $\mathcal{G}, q \not\models^T (Q_i x_i)_{1\le i\le l}.\ \Gamma_{d,s,h}$ for any $h \succ_s b_{q,d,s}$.

This concludes the inductive definition of $\Gamma_{d,s,b_{q,d,s}}$. We now prove that it satisfies the following lemma:

Lemma 9 *For any $q \in Q$, any $d \in \mathsf{VS}$ and any $s \in \{0,1\}^n$, it holds*

$$\mathcal{G}, q \models (Q_i x_i)_{1\le i\le l}.\ \Gamma_{d,s,b_{q,d,s}} \tag{9}$$

$$\mathcal{G}, q \models^T (\overline{Q}_i x_i)_{1\le i\le l}.\ \neg\Gamma_{d,s,h} \qquad \text{for any } h \succ_s b_{q,d,s}. \tag{10}$$

Proof The first result is a direct consequence of the construction: the values for $b_{q,d,s}$ have been selected so that $\mathcal{G}, q \models^T (Q_i x_i)_{1\le i\le l}.\ \Gamma_{d,s,b_{q,d,s}}$.

To prove the second part, we again turn the satisfaction of $\Gamma_{d,s,h}$, for $h \succ_s b_{q,d,s}$, into a parity game, as for the proof of Prop. 2. We only sketch the proof here, as it involves the same ingredients.

The parity game is obtained from $\mathcal{G}$ by replacing each state by a quantification game. We also introduce two sink states, q_{even} and q_{odd}, which are winning for Player $P_\exists$ and for Player $P_\forall$ respectively. When arriving at a leaf $(q, d, \mathfrak{m})$ of the (q, d)-copy of the quantification game, there may be one of the following three transitions available:

- if there is a state q' such that for all j with $s(j) = 1$, it holds $q' = \Delta(q, m_{\beta_j})$ (in other terms, the moves selected in the current quantification game generate the same transition for all the goals enabled by s), then there is a single transition to (q', d', ϵ), where $d' = \mathsf{succ}(d, q')$.
- otherwise, if there is an extended partition $\tau = (s_\kappa, q_\kappa, d_\kappa)_{1\le\kappa\le\lambda}$ of s such that $c_{s,\tau} \succeq_s h$ and, for all $1\le\kappa\le\lambda$, for all j such that $s_\kappa(j) = 1$, we have $\Delta(q, m_{\beta_j}) = q_\kappa$ and $\mathsf{succ}(d, q_\kappa) = d_\kappa$, then there is a transition from $(q, d, \mathfrak{m})$ to q_{even}.
- otherwise, there is a transition from $(q, d, \mathfrak{m})$ to q_{odd}.

The priorities defining the parity condition are inherited from those in $D_{s,h}$.

Since $\mathcal{G}, q \not\models^T (Q_i x_i)_{1\le i\le l}.\ \Gamma_{d,s,h}$, Player $P_\exists$ does not have a winning strategy in this game, and by determinacy Player $P_\forall$ has one. From the winning strategy of Player $P_\forall$, we obtain a timeline map $\overline{\vartheta}_{q,d,s,h}$ for $(\overline{Q}_i x_i)_{1\le i\le l}$ witnessing the fact that $\mathcal{G}, q \models^T (\overline{Q}_i x_i)_{1\le i\le l}.\ \neg\Gamma_{d,s,h}$. □

Remark 2 While the definition of $\Gamma_{d,s,b_{q,d,s}}$ (and in particular of $b_{q,d,s}$) is not effective, the parity games defined in the proof above can be used to compute each

$b_{q,d,s}$ and $\Gamma_{d,s,b_{q,d,s}}$. Indeed, such parity games can be used to decide whether $\mathcal{G}, q \models^T (Q_i x_i)_{1\le i\le l}.\ \Gamma_{d,s,h}$ for all h, and selecting a maximal value for which the result holds. Then $b_{q_0,d_0,\mathbf{1}} \in F^n$ implies $\mathcal{G}, q_0 \models^T (Q_i x_i)_{1\le i\le l} F^n(\beta_j.\ \varphi_j)_{1\le j\le n}$.

Each parity game has size doubly-exponential, with exponentially-many priorities; hence they can be solved in 2-EXPTIME. The number of games to solve is also doubly-exponential, so that the whole algorithm runs in 2-EXPTIME.

Applying Lemma 9, we fix a timeline map $\vartheta_{q,d,s}$ for $(Q_i x_i)_{1\le i\le l}$ witnessing (9), and for each $h \succ_s b_{q,d,s}$, a timeline map $\overline{\vartheta}_{q,d,s,h}$ for $(\overline{Q}_i x_i)_{1\le i\le l}$ witnessing (10).

We now focus on the operator obtained at the end of the induction, when $s = \mathbf{1}$. Following Prop. 8, $\mathcal{L}(D_{\mathbf{1},f})$ does not depend on the exact value of f, as soon as it is in F^n. We then let

$$\Gamma_{F^n} = \Gamma^{\mathsf{stick}}_{d_0,\mathbf{1},f} \vee \Gamma^{\mathsf{sep}}_{d_0,\mathbf{1},\Upsilon_{F^n}}$$

where f is any element of F^n (remember F^n is assumed to be non-empty), d_0 is the vector of initial states of the automata in $\mathcal{D}$, and $\Upsilon_{F^n} = \{\mathsf{Part}(\mathbf{1}) \mid c_{\mathbf{1},\tau} \in F^n\}$.

We write $\vartheta_{\mathbf{1}}$ and $\overline{\vartheta}_{\mathbf{1}}$ for the maps $\vartheta_{q_0,d_0,\mathbf{1}}$ and $\overline{\vartheta}_{q_0,d_0,\mathbf{1},h}$ for some $h \in F^n$, as given by Lemma 9. From the discussion above, $\overline{\vartheta}_{q_0,d_0,\mathbf{1},h}$ does not depend on the choice of h in F^n, and we simply write it $\overline{\vartheta}_{q_0,d_0,\mathbf{1}}$.

Then:

Lemma 10 *If $\mathcal{G}, q_0 \models^T (Q_i x_i)_{1\le i\le l}$, then $\vartheta_{\mathbf{1}}$ witnesses the fact that $\mathcal{G}, q_0 \models^T (Q_i x_i)_{1\le i\le l}.\ \Gamma_{F^n}$. Conversely, if $\mathcal{G}, q_0 \not\models^T (Q_i x_i)_{1\le i\le l}.\ \Gamma_{F^n}$, then $\overline{\vartheta}_{\mathbf{1}}$ witness the fact that $\mathcal{G}, q_0 \models^T (\overline{Q}_i x_i)_{1\le i\le l}.\ \neg\Gamma_{F^n}$.*

Proof The first part directly follows from the previous lemma. For the second part, $\mathcal{G}, q_0 \not\models^T (Q_i x_i)_{1\le i\le l}.\ \Gamma_{F^n}$ means that $b_{q_0,d_0,\mathbf{1}} \notin F^n$. Hence for any $f \in F^n$, we have $f \succ_s b_{q_0,d_0,\mathbf{1}}$, so that $\overline{\vartheta}_{q_0,d_0\mathbf{1}}$ is a witness that $\mathcal{G}, q \models^T (\overline{Q}_i x_i)_{1\le i\le l}.\ \neg\Gamma_{F^n}$. □

5.5.4 Compiling optimal maps

From Lemma 9, we have timeline maps for each q, d and s. We now compile them into two map θ and $\overline{\theta}$. The construction is inductive, along histories.

Pick a history ρ starting from q_0 and strategies for universally-quantified variables $w\colon \mathcal{V}^\forall \to (\mathsf{Hist} \to \mathsf{Act})$. Assuming θ has been defined along all strict prefixes of ρ, a goal $\beta_j.\ \varphi_j$ is said *active* after ρ w.r.t. $\theta(w)$ if the following condition holds:

$$\forall i < |\rho|.\ \rho(i+1) = \Delta(\rho(i), (\theta(w)(\beta_j(A))(\rho_{\le i}))_{A\in\mathsf{Agt}}).$$

In other terms, $\beta_j.\ \varphi_j$ is active after ρ w.r.t. $\theta(w)$ if ρ is the outcome of strategies prescribed by $\theta(w)$ under assignment β_j. We let $s_{\rho,\theta(w)}$ be the element of $\{0,1\}^n$ such that $s_{\rho,\theta(w)}(j) = 1$ if, and only if, $\beta_j.\ \varphi_j$ is active after ρ w.r.t. $\theta(w)$.

We now define $\theta(w)(x_i)(\rho)$ for all $x_i \in \mathcal{V}$:

- if $x_i \in \mathcal{V}^\forall$, we let $\theta(w)(x_i)(\rho) = w(x_i)(\rho)$;
- if $x_i \in \mathcal{V}^\exists$, we consider two cases:
  - if $s_{\rho,\theta(w)} = 1$, then all goals are still active, and θ follows the map $\vartheta_{\mathbf{1}}$: $\theta(w)(x_i)(\rho) = \vartheta_{\mathbf{1}}(w)(x_i)(\rho)$.

- otherwise, we let ρ_1 be the maximal prefix of ρ for which $s_{\rho_1,\theta(w)} \neq s_{\rho,\theta(w)}$. We may then write $\rho = \rho_1 \cdot \rho_2$, and let $q_1 = \mathsf{last}(\rho_1)$ and $d_1 = \mathsf{succ}(d_0, \rho_1)$. We then let $\theta(w)(x_i)(\rho) = \vartheta_{q_1,d_1,s_{\rho,\theta(w)}}(w_{\overrightarrow{\rho_1}})(x_i)(\rho_2)$.

The dual map $\overline{\theta}$ is defined in the same way, using maps $\overline{\vartheta}$ in place of ϑ.

The following result will conclude our proof of Theorem 1.

Lemma 11 *There exists a context χ with domain $\mathcal{V}$ such that $\theta(\chi_{|\mathcal{V}^\forall}) = \chi$ and $\overline{\theta}(\chi_{|\mathcal{V}^\exists}) = \chi$. It satisfies*

$$\begin{aligned} \mathcal{G}, q_0 \models_\chi \Gamma_{F^n} \quad &\Rightarrow \quad \forall w \in (\mathit{Hist}_\mathcal{G} \to \mathit{Act})^{\mathcal{V}^\forall}. \mathcal{G}, q_0 \models_{\theta(w)} F^n(\beta_j.\, \varphi_j)_{1\leq j\leq n} \\ \mathcal{G}, q_0 \models_\chi \neg\Gamma_{F^n} \quad &\Rightarrow \quad \forall \overline{w} \in (\mathit{Hist}_\mathcal{G} \to \mathit{Act})^{\mathcal{V}^\exists}. \mathcal{G}, q_0 \models_{\overline{\theta}(\overline{w})} \overline{F^n}(\beta_j.\, \varphi_j)_{1\leq j\leq n} \end{aligned}$$

Proof We use the same technique as in the proof of Prop. 5: from θ and $\overline{\theta}$, we build a strategy context χ on $\mathcal{V}$ such that $\theta(\chi_{|\mathcal{V}^\forall}) = \chi$ and $\overline{\theta}(\chi_{|\mathcal{V}^\exists}) = \chi$.

We introduce some more notations. For $w \colon \mathcal{V}^\forall \to (\mathsf{Hist}_\mathcal{G} \to \mathsf{Act})$, we let

- π_j^w be the outcome $\mathsf{out}(q_0, (\theta(w)((\beta_j(A))_{A\in\mathsf{Agt}}))$ for all $1 \leq j \leq n$;
- f^w be the element of $\{0,1\}^n$ such that $f^w(j) = 1$ if, and only if, $\pi_j^w \models \varphi_j$;
- $R^w \subseteq \{0,1\}^n \times \mathsf{Hist}_\mathcal{G}$ be the relation such that $(s,\rho) \in R^w$ if, and only if, $s = s_{\rho,\theta(w)}$ and ρ is minimal (meaning for any strict prefix ρ' of ρ, it holds $(s,\rho') \notin R^w$).

Lemma 12 *For any $w \colon \mathcal{V}^\forall \to (\mathit{Hist}_\mathcal{G} \to \mathit{Act})$ and any $\rho \in \mathit{Hist}$, letting $d_\rho = \mathit{succ}(d_0, \rho)$, it holds*

$$\forall s \in \{0,1\}^n.\, (s,\rho) \in R^w \Rightarrow b_{\mathit{last}(\rho),d_\rho,s} \preceq_s f^w.$$

Proof Fix some $w \in (\mathsf{Hist}_\mathcal{G} \to \mathsf{Act})^{\mathcal{V}^\forall}$. The proof proceeds by induction on $|s|$.

Base case: $|s| = 1$ Assume $(s,\rho) \in R^w$. As $|s| = 1$, there is a unique goal, say $\beta_{j_0}.\, \varphi_{j_0}$, active along ρ w.r.t. $\theta(w)$. By definition of θ, $\pi_{j_0} = \rho \cdot \eta$ where η is the outcome of $\vartheta_{\mathsf{last}(\rho),d_\rho,s}(w_{\overrightarrow{\rho}})((\beta_j(A))_{A\in\mathsf{Agt}})$ from $\mathsf{last}(\rho)$.

Because $|s| = 1$, we have $\Gamma_{d_\rho,s,b_{\mathsf{last}(\rho),d_\rho,s}} = \Gamma^{\mathsf{stick}}_{d_\rho,s,b_{\mathsf{last}(\rho),d_\rho,s}}$. The map $\vartheta_{\mathsf{last}(\rho),d_\rho,s}$ is a witness that $\mathcal{G}, \mathsf{last}(\rho) \mathrel{|\!\!\!\equiv}^T (Q_i x_i)_{1\leq i\leq l}\Gamma_{d_\rho,s,b_{\mathsf{last}\rho,d_\rho,s}}$; therefore it also witnesses that $\mathcal{G}, \mathsf{last}(\rho) \mathrel{|\!\!\!\equiv}^T (Q_i x_i)_{1\leq i\leq l}\Gamma^{\mathsf{stick}}_{d_\rho,s,b_{\mathsf{last}(\rho),d_\rho,s}}$. By definition of the Γ^{stick} operators, this implies that for any w, the outcome of $\vartheta_{\mathsf{last}(\rho),d_\rho,s}(w_{\overrightarrow{\rho}})$ from $\mathsf{last}(\rho)$ is accepted by the automaton $D^{d_\rho}_{s,b_{\mathsf{last}(\rho),d_\rho,s}}$; in particular, η is accepted by $D^{d_\rho}_{s,b_{\mathsf{last}(\rho),d_\rho,s}}$.

The automaton $D^{d_\rho}_{s,b_{\mathsf{last}(\rho),d_\rho,s}}$ accepts paths which give better results (w.r.t. $\preceq_s$) for the objectives $(\beta_j.\, \varphi_j)_{j|s(j)=1}$ than $b_{\mathsf{last}(\rho),d_\rho,s}$. In other terms, we have $b_{\mathsf{last}(\rho),d_\rho,s} \preceq_s f^w$.

Induction step We assume that the Proposition 12 holds for any elements $s \in \{0, 1\}^n$ of size $|s| < \alpha$. We now consider for the induction step an element $s \in \{0, 1\}^n$ such that $|s| = \alpha$ and $(s, \rho) \in R^w$.

- if the enabled goals all follow the same outcome, i.e., if there exists an infinite path η such that $\pi_j = \rho \cdot \eta$ for all j having $s(j) = 1$, then with arguments similar to those of the base case, we get $b_{\mathsf{last}(\rho),d_\rho,s} \preceq_s f^w$.
- otherwise, the goals enabled by s split following an extended partition $\tau = (s_\kappa, q_\kappa, d_\kappa)_{\kappa \leq \lambda}$. We let η be the history from the last state of ρ to the point where the goals split.

  The map $\vartheta_{\mathsf{last}(\rho),d_\rho,s}$ witnesses that $\mathcal{G}, \mathsf{last}(\rho) \models^T \Gamma_{d,s,b_{\mathsf{last}(\rho),d_\rho,s}}$; therefore η may only reach a partition τ such that

  $$b_{\mathsf{last}(\rho),d_\rho,s} \preceq_s c_{s,\tau} \tag{11}$$

  This partition τ is such that for any $1 \leq \kappa \leq \lambda$, it holds $(s_\kappa, \rho \cdot \eta \cdot q_\kappa) \in R^w$; using the induction hypothesis, we get

  $$s_\kappa \curlywedge b_{q_\kappa,d_\kappa,s_\kappa} \preceq_{s_\kappa} f^w \tag{12}$$

  Then, using Lemma 7 repeatedly on the $(s_\kappa)_{1 \leq \kappa \leq \lambda}$, and (12), we obtain

  $$\begin{aligned} s_1 \curlywedge b_{q_1,d_1,s_1} \preceq_{s_1} f^w &\Rightarrow (s_1 \curlywedge b_{q_1,d_1,s_1}) \curlyvee (s_2 \curlywedge b_{q_2,d_2,s_2}) \preceq_{s_1 \curlyvee s_2} f^w \\ &\Rightarrow \ldots \\ &\Rightarrow (s_1 \curlywedge b_{q_1,d_1,s_1}) \curlyvee \cdots \curlyvee (s_\lambda \curlywedge b_{q_\lambda,d_\lambda,s_\lambda}) \preceq_{s_1 \curlyvee \cdots \curlyvee s_\lambda} f^w \\ &\Rightarrow c_{s,\tau} \preceq_s f^w. \end{aligned}$$

  Combined with (11), we get $b_{\mathsf{last}(\rho),d_\rho,s} \preceq_s c_{s,\tau} \preceq_s f^w$.

□

Lemma 13 *$\mathcal{G}, q_0 \models_\chi \Gamma_{F^n}$ if, and only if, $b_{q_0,d_0,\mathbf{1}} \in F^n$.*

Proof Assume that $b_{q_0,d_0,\mathbf{1}} \in \overline{F^n}$. Then $\mathcal{G}, q_0 \not\models^T (Q_i x_i)_{1 \leq i \leq l}.\, \Gamma_{F^n}$. Applying Lemma 10, the map $\overline{\vartheta_\mathbf{1}}$ (and therefore $\overline{\theta}$, which act as $\overline{\vartheta_\mathbf{1}}$ before goals branch along different paths) witnesses $\mathcal{G}, q_0 \not\models^T (Q_i x_i)_{1 \leq i \leq l}.\, \Gamma_{F^n}$. This implies that $\mathcal{G}, q_0 \not\models_\chi \Gamma_{F^n}$, which contradicts the hypothesis. □

Conversely, if $b_{q_0,d_0,\mathbf{1}} \in F^n$, then $\mathcal{G}, q_0 \models^T (Q_i x_i)_{1 \leq i \leq l}.\, \Gamma_{F^n}$, which is witnessed by map ϑ_1. Thus $\mathcal{G}, q_0 \models_\chi \Gamma_{F^n}$.

We are now ready to prove the first part of Lemma 11: consider a function $w \colon \mathcal{V}^\forall \to (\mathsf{Hist}_\mathcal{G} \to \mathsf{Act})$. By Lemma 12 applied to w, $s = \mathbf{1}$, and $\rho = q_0$, we get that $b_{q_0,d_0,\mathbf{1}} \preceq_\mathbf{1} f^w$. Now, by Lemma 13, $b_{q_0,d_0,\mathbf{1}} \in F^n$, therefore the element f^w, being greater than $b_{q_0,d_0,\mathbf{1}}$ for $\preceq_\mathbf{1}$, must also be in F^n, which means that $\mathcal{G}, q_0 \models_{\theta(w)} F^n(\beta_j.\, \varphi_j)_{1 \leq j \leq n}$.

The second implication of the lemma is proven using similar arguments.

Lemma 11 allows us to conclude that at least one of φ and $\neg\varphi$ must hold on $\mathcal{G}$ for $\models^T$. Lemma 5 implies that at most one can hold. Combining both we get that exactly one holds.

From this proof, we get:

Corollary 3 *Model checking SL[EG] for $\models^T$ is 2-EXPTIME-complete.*

Remark 3 Notice that we do not get the twin of Corollary 1 here, and actually $\models^T$ and $\models^C$ differ over SL[EG]$^\flat$. Indeed, the proof of Prop. 4 provides a counterexample:

- as shown in the proof of Prop. 4, the game $\mathcal{G}$ and formula $\varphi \in$ SL[CG]$^\flat$ of Fig. 6 are such that $\mathcal{G}, q_0 \models^T \varphi$;
- considering the classical semantics, because of the conjunction of goals, any strategy for y for which the rest of the formula is fulfilled must play differently in states q_1 and q_2. On the other hand, in order to fulfill the first conjunct for any strategy x_A, then the strategy y must play to p_1 from both q_1 and q_2. Hence no such strategy exist.

5.6 Maximality of SL[EG]$^\flat$

Finally, we prove that SL[EG]$^\flat$ is, in a sense, maximal for the first property of Theorem 1:

Proposition 9 *For any non-semi-stable boolean set $F^n \subseteq \{0,1\}^n$, there exists a SL[BG]$^\flat$ 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 $\mathcal{G}, q_0 \not\models^T \varphi$.*

Proof We consider again the game $\mathcal{G}$ depicted on Fig. 6, with two agents 🟥 and 🟠. Let F^n be a non-semi-stable set over $\{0,1\}^n$. Then there must exist $f_1, f_2 \in F^n$, and $s \in \{0,1\}^n$, such that $(f_1 \curlywedge s) \curlyvee (f_2 \curlywedge \overline{s}) \notin F^n$ and $(f_2 \curlywedge s) \curlyvee (f_1 \curlywedge \overline{s}) \notin F^n$. We then let

$$\varphi = \forall y_1.\, \forall y_2.\, \forall x_1.\, \exists x_2.\, F^n(\beta_1.\, \varphi_1, \ldots, \beta_n.\, \varphi_n)$$

where

$$\beta_i = \begin{cases} \mathsf{assign}(\text{🟥} \mapsto y_1;\ \text{🟠} \mapsto x_1) & \text{if } s(i) = 1 \\ \mathsf{assign}(\text{🟥} \mapsto y_2;\ \text{🟠} \mapsto x_2) & \text{if } s(i) = 0 \end{cases}$$

and

$$\varphi_i = \begin{cases} \mathbf{F}\, p_1 \vee \mathbf{F}\, p_2 & \text{if } f_1(i) = f_2(i) = 1 \\ \mathbf{F}\, p_1 & \text{if } f_1(i) = 1 \text{ and } f_2(i) = 0 \\ \mathbf{F}\, p_2 & \text{if } f_1(i) = 0 \text{ and } f_2(i) = 1 \\ \mathtt{false} & \text{if } f_1(i) = f_2(i) = 0 \end{cases}$$

Formulas φ_i have been built to satisfy the following property:

Lemma 14 *Let ρ be a maximal run of $\mathcal{G}$ from q_0. Let $k \in \{1,2\}$ be such that ρ visits a state labelled with p_k. Then for any $1 \leq i \leq n$, we have $\rho \models \varphi_i$ if, and only if, $f_k(i) = 1$.*

The following two lemmas conclude the proof:

Lemma 15 $\mathcal{G}, q_0 \not\models^T \varphi$.

Proof Towards a contradiction, assume that $\mathcal{G}, q_0 \models^T \varphi$. Let θ be a timeline map witnessing this fact. We let σ_1 (resp. σ_2) be the strategy that maps history q_0 to q_1 (resp. q_2). We let τ_1 be such that $\tau_1(q_0 \cdot q_1) = p_1$. This defines a valuation w, respectively mapping y_1, y_2 and x_1 to σ_1, σ_2 and τ_1. Then the strategy $\tau_2 = \theta(w)(x_2)$ is such that

$$\mathcal{G}, q_0 \models_{\theta(w)} F^n(\beta_1.\, \varphi_1, \ldots, \beta_n.\, \varphi_n).$$

Now, consider the valuation w' obtained from w by changing the strategy for x_1 to τ'_1, where $\tau'_1(q_0 \cdot q_1) = p_2$. Then $\theta(w')(x_2) = \theta(w)(x_2) = \tau_2$, since θ is a timeline map. Since θ witnesses the satisfaction of φ, we also have

$$\mathcal{G}, q_0 \models_{\theta(w')} F^n(\beta_1.\, \varphi_1, \ldots, \beta_n.\, \varphi_n).$$

Let v and v' be the vectors in $\{0, 1\}^n$ representing the values of the goals $(\beta_1.\, \varphi_1, \ldots, \beta_n.\, \varphi_n)$ under $\theta(w)$ and $\theta(w')$, respectively. Then v and v' are in F^n. However:

- if $\tau_2(q_0 \cdot q_2) = p_1$, then under $\theta(w')$, for any $1 \leq i \leq n$:
  - if $s_i = 1$, strategies σ_1 and τ'_1 are applied, so that the game ends in p_2; then $v'_i = 1$ if, and only if, $f_2(i) = 1$;
  - if $s_i = 0$, strategies σ_2 and τ_2 are used, and the game goes to p_1; then $v'_i = 1$ if, and only if, $f_1(i) = 1$.

  In the end, we have $v' = (f_1 \curlywedge \overline{s}) \curlyvee (f_2 \curlywedge s)$, which is not in F^n.
- if $\tau_2(q_0 \cdot q_2) = p_2$, then under $\theta(w)$, for any $1 \leq i \leq n$:
  - if $s_i = 1$, strategies σ_1 and τ_1 are applied, so that the game ends in p_1; then $v_i = 1$ if, and only if, $f_1(i) = 1$;
  - if $s_i = 0$, strategies σ_2 and τ_2 are used, and the game goes to p_2; then $v_i = 1$ if, and only if, $f_2(i) = 1$.

  In the end, we have $v = (f_1 \curlywedge s) \curlyvee (f_2 \curlywedge \overline{s})$, which also is not in F^n.

Both cases lead to a contradiction, so that our hypothesis that $\mathcal{G}, q_0 \models^T \varphi$ can only be wrong. □

Lemma 16 $\mathcal{G}, q_0 \not\models^T \neg\varphi$.

Proof We use similar arguments as above: we assume $\mathcal{G}, q_0 \models^T \neg\varphi$, and fix a witnessing timeline map $\overline{\theta}$ for $\neg\varphi$.

We consider four valuations w^{11}, w^{12}, w^{21} and w^{22} for x_2, such that $w^{jk}(x_2)(\rho) = w^{j'k'}(x_2)(q_0)$ (the exact value is not important) and $w^{jk}(x_2)(q_0 \cdot q_1) = p_i$ and $w^{jk}(x_2)(q_0 \cdot q_2) = p_j$. We let $\sigma_1 = \overline{\theta}(w^{jk})(y_1)$, $\sigma_2 = \overline{\theta}(w^{jk})(y_2)$ and $\tau_1 = \overline{\theta}(w^{jk})(x_1)$. Notice that those strategies do not depend on i and j, since $\overline{\theta}$ is a timeline map for $\neg\varphi$. We write v_i^{jk} for the vector representing the truth value of goal $\beta_i.\, \varphi_i$ under valuation $\overline{\theta}(w^{jk})$.

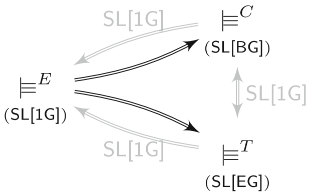


Fig. 10 Relations between classical, elementary and timeline semantics

Assume that $\sigma_2(q_0) = q_1$, and that $\tau_1(q_0 \cdot \sigma_1(q_0)) = p_1$. Then under w^{11} (i.e., when $\tau_2(q_0 \cdot q_1) = p1$), for any $1 \leq i \leq n$, the outcome of strategy assignment β_i from q_0 goes to p_1. Hence $v^{11} = f_1$, which is in F^n, contradicting the fact that $\overline{\theta}$ witnesses $\mathcal{G}, q_0 \models^T \neg\varphi$. Similar arguments apply if $\tau_1(q_0 \cdot \sigma_1(q_0)) = p_2$, and when $\sigma_2(q_0) = q_2$. Thus our assumption that $\mathcal{G}, q_0 \models^T \neg\varphi$ cannot be correct. □

6 Conclusions and Future Works

In this paper, we have studied various semantics of SL, depending on how the successive strategy quantifiers in an SL formula may depend on each other. Following [30], we defined a natural translation of the elementary semantics of SL[1G] into a two-player turn-based parity game, and introduced a new *timeline semantics* for SL[BG] that better corresponds to this translation. For this new semantics, we defined a fragment SL[EG] for which the timeline semantics can be model-checked in 2-EXPTIME. Figure 10 represents the relations between those semantics (with implications in grey only valid for SL[1G]), as well as the maximal fragments of SL[BG] for which the semantical and syntactical negations coincide.

While our work clarifies the setting of strategy dependences in SL, those various semantics of SL remains to be fully understood, in particular as to which situations are better suited for which semantics. Of course, studying the decidability and complexity of model checking for the different semantics and fragments of SL[BG] is a natural continuation of this work. Studying quantitative or epistemic extensions of SL[EG] under the timeline semantics is also a natural direction to follow. Finally, since our approach relies on translations to two-player parity games, our model-checking algorithm would be a good candidate for being implemented e.g. in the tool MCMAS.

References

1. Ågotnes, T., Goranko, V., Jamroga, W.: Alternating-time temporal logics with irrevocable strategies. In: Samet, D. (ed.) Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge (TARK'07), pp. 15–24 (2007)
2. Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. J. ACM **49**(5), 672–713 (2002). https://doi.org/10.1145/585265.585270
3. Aminof, B., Malvone, V., Murano, A., Rubin, S.: Graded modalities in strategy logic. Inf. Comput. **261**(4), 634–649 (2018). https://doi.org/10.1016/j.ic.2018.02.021

4. Berthon, R., Maubert, B., Murano, A., Rubin, S., Vardi, M.Y.: Strategy logic with imperfect information. In: Proceedings of the 32th Annual Symposium on Logic in Computer Science (LICS'17), pp. 1–12. IEEE Comp. Soc. Press. https://doi.org/10.1109/LICS.2017.8005136 (2017)
5. Bouyer, P., Gardy, P., Markey, N.: Weighted strategy logic with boolean goals over one-counter games. In: Harsha, P., Ramalingam, G. (eds.) Proceedings of the 35th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'15), volume 45 of Leibniz International Proceedings in Informatics, Leibniz-Zentrum für Informatik, pp. 69–83 (2015). https://doi.org/10.4230/LIPIcs.FSTTCS.2015.69
6. Bouyer, P., Gardy, P., Markey, N.: On the semantics of strategy logic. Inf. Process. Lett. **116**(2), 75–79 (2016). https://doi.org/10.1016/j.ipl.2015.10.004
7. Brenguier, R., Raskin, J.-F., Sankur, O.: Assume-admissible synthesis. Acta Inform. **54**(1), 41–83 (2017). https://doi.org/10.1007/s00236-016-0273-2
8. Brihaye, T., Da Costa, A., Laroussinie, F., Markey, N.: ATL with strategy contexts and bounded memory. In: Artemov, S.N., Nerode, A. (eds.) Proceedings of the International Symposium Logical Foundations of Computer Science (LFCS'09), volume 5407 of Lecture Notes in Computer Science, pp. 92–106. Springer (2009). https://doi.org/10.1007/978-3-540-92687-0_7
9. Čermák, P., Lomuscio, A., Mogavero, F., Murano, A.: MCMAS-SLK: A model checker for the verification of strategy logic specifications. In: Biere, A., Bloem, R. (eds.) Proceedings of the 26th International Conference on Computer Aided Verification (CAV'14), volume 8559 of Lecture Notes in Computer Science, pp. 525–532. Springer (2014). https://doi.org/10.1007/978-3-319-08867-9_34
10. Čermák, P., Lomuscio, A., Murano, A.: Verifying and synthesising multi-agent systems against one-goal strategy logic specifications. In: Bonet, B., Koenig, S. (eds.) Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15), pp. 2038–2044. AAAI Press (2015)
11. Chatterjee, K., Henzinger, T.A., Piterman, N.: Strategy logic. In: Caires, L., Vasconcelos, V.T. (eds.) Proceedings of the 18th International Conference on Concurrency Theory (CONCUR'07), volume 4703 of Lecture Notes in Computer Science, pp. 59–73. Springer (2007). https://doi.org/10.1007/978-3-540-74407-8_5
12. Clarke, E.M., Emerson, E.A.: Design and synthesis of synchronization skeletons using branching-time temporal logic. In: Kozen, D.C. (ed.) Proceedings of the 3rd Workshop on Logics of Programs (LOP'81), volume 131 of Lecture Notes in Computer Science, pp. 52–71. Springer (1982). https://doi.org/10.1007/BFb0025774
13. Clarke, E.M., Grumberg, O., Peled, D.A.: Model checking. MIT Press, New York (2000)
14. Condurache, R., Filiot, E., Gentilini, R., Raskin, J.-F., Sangiorgi, D.: The complexity of rational synthesis. In: Chatzigiannakis, I., Mitzenmacher, M., Rabani, Y. (eds.) Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP'16) – Part II, volume 55 of Leibniz International Proceedings in Informatics, pp. 121:1–121:15. Leibniz-Zentrum Für Informatik (2016). https://doi.org/10.4230/LIPIcs.ICALP.2016.121
15. Dima, C., Ţiplea, F.L.: Model-checking ATL under imperfect information and perfect recall semantics is undecidable. Research Report arXiv:1102.4225 (2011)
16. Emerson, E.A., Jutla, C.S.: Tree automata, mu-calculus and determinacy. In: Proceedings of the 32nd Annual Symposium on Foundations of Computer Science (FOCS'91), pp. 368–377. IEEE Comp. Soc. Press. https://doi.org/10.1109/SFCS.1991.185392 (1991)
17. Fijalkow, N., Maubert, B., Murano, A., Rubin, S.: Quantifying bounds in strategy logic. In: Ghica, D.R., Jung, A. (eds.) Proceedings of the 27th EACSL Annual Conference on Computer Science Logic (CSL'18), volume 119 of Leibniz International Proceedings in Informatics, pp. 23:1–23:23. Leibniz-Zentrum Für Informatik (2018). https://doi.org/10.4230/LIPIcs.CSL.2018.23
18. Fisman, D., Kupferman, O., Lustig, Y.: Rational synthesis. In: Esparza, J., Majumdar, R. (eds.) 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, pp. 190–204. Springer (2010). https://doi.org/10.1007/978-3-642-12002-2_16
19. Gardy, P., Bouyer, P., Markey, N.: Dependences in strategy logic. In: Proceedings of the 35th Annual Symposium on Theoretical Aspects of Computer Science (STACS'18), volume 96 of Leibniz International Proceedings in Informatics, pp. 34:1–34:15. Leibniz-Zentrum Für Informatik (2018). https://doi.org/10.4230/LIPIcs.STACS.2018.34
20. Goranko, V., van Drimmelen, G.: Complete axiomatization and decidability of alternating-time temporal logic. Theor. Comput. Sci. **353**(1-3), 93–117 (2006)
21. Guelev, D.P., Dima, C.: Epistemic ATL with perfect recall, past and strategy contexts. In: Fisher, M., van der Torre, L.W.N., Dastani, M., Governatori, G. (eds.) Proceedings of the 13th International

Workshop on Computational Logic in Multi-Agent Systems (CLIMA'12), volume 7486 of Lecture Notes in Artificial Intelligence, pp. 77–93. Springer (2012). https://doi.org/10.1007/978-3-642-32897-8_7

22. Gutierrez, J., Harrenstein, P., Perelli, G., Wooldridge, M.: Nash equilibrium and bisimulation invariance. In: Meyer, R., Nestmann, U. (eds.) Proceedings of the 28th International Conference on Concurrency Theory (CONCUR'17), volume 85 of Leibniz International Proceedings in Informatics, pp. 17:1–17:16. Leibniz-Zentrum Für Informatik (2017). https://doi.org/10.4230/LIPIcs.CONCUR.2017.17
23. Henkin, L.: Some remarks on infinitely long formulas. In: Infinitistic Methods – Proceedings of the Symposium on Foundations of Mathematics, pp. 167–183. Pergamon Press (1961)
24. Hintikka, J., Sandu, G.: Informational independence as a semantical phenomenon. In: Fenstad, J.E., Frolov, I.T., Hilpinen, R. (eds.) Proceedings of the 8th International Congress of Logic, Methodology and Philosophy of Science, volume 70 of Studies in Logic and the Foundations of Mathematics, pp. 571–589, North-Holland (1989). https://doi.org/10.1016/S0049-237X(08)70066-1
25. Huang, X., van der Meyden, R.: An epistemic strategy logic. ACM Trans. Comput. Log. **19**(4), 26:1–26:45 (2018). https://doi.org/10.1145/3233769
26. Kupferman, O., Perelli, G., Vardi, M.Y.: Synthesis with rational environments. Ann. Math. Artif. Intell. **78**(1), 3–20 (2016). https://doi.org/10.1007/s10472-016-9508-8
27. Laroussinie, F., Markey, N.: Augmenting ATL with strategy contexts. Inf. Comput. **245**, 98–123 (2015). https://doi.org/10.1016/j.ic.2014.12.020
28. Laroussinie, F., Markey, N., Oreiby, G.: On the expressiveness and complexity of ATL. Logical Methods in Computer Science **4**(2:7), 1–25 (2008). https://doi.org/10.2168/LMCS-4(2:7)2008
29. Mogavero, F., Murano, A., Perelli, G., Vardi, M.Y.: What makes ATL* decidable? A decidable fragment of strategy logic. In: Koutny, M., Ulidowski, I. (eds.) Proceedings of the 23rd International Conference on Concurrency Theory (CONCUR'12), volume 7454 of Lecture Notes in Computer Science, pp. 193–208. Springer (2012)
30. Mogavero, F., Murano, A., Perelli, G., Vardi, M.Y.: Reasoning about strategies: On the model-checking problem. ACM Trans. Comput. Log. **15**(4), 34:1–34:47 (2014). https://doi.org/10.1145/2631917
31. Mogavero, F., Murano, A., Perelli, G., Vardi, M.Y.: Reasoning about strategies: On the satisfiability problem. Logical Methods in Computer Science **13**(1:9), 1–37 (2017). https://doi.org/10.23638/LMCS-13(1:9)2017
32. Mogavero, F., Murano, A., Sauro, L.: 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 (2013)
33. Mogavero, F., Murano, A., Sauro, L.: A behavioral hierarchy of strategy logic. In: Bulling, N., van der Torre, L.W.N., Villata, S., Jamroga, W., Vasconcelos, W.W. (eds.) Proceedings of the 15th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA'14), volume 8624 of Lecture Notes in Artificial Intelligence, pp. 148–165. Springer (2014). https://doi.org/10.1007/978-3-319-09764-0_10
34. Mogavero, F., Murano, A., Vardi, M.Y.: Reasoning about strategies. In: Lodaya, K., Mahajan, M. (eds.) Proceedings of the 30th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'10), volume 8 of Leibniz International Proceedings in Informatics, pp. 133–144. Leibniz-Zentrum Für Informatik (2010). https://doi.org/10.4230/LIPIcs.FSTTCS.2010.133
35. Mostowski, A.: Games with forbidden positions. Research report 78 university of Danzig (1991)
36. Pnueli, A.: The temporal logic of programs. In: Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS'77), pp. 46–57. IEEE Comp. Soc. Press. https://doi.org/10.1109/SFCS.1977.32 (1977)
37. Queille, J.-P., Sifakis, J.: Specification and verification of concurrent systems in CESAR. In: Dezani-Ciancaglini, M., Montanari, U. (eds.) Proceedings of the 5th International Symposium on Programming (SOP'82), volume 137 of Lecture Notes in Computer Science, pp. 337–351. Springer (1982). https://doi.org/10.1007/3-540-11494-7_22
38. Väänänen, J.: Dependence Logic: A New Approach to Independence-Friendly Logic, volume 70 of London Mathematical Society Student Texts. Cambridge University Press, Cambridge (2007). https://doi.org/10.1017/CBO9780511611193

39. Wang, F., Huang, C.-H., Yu, F.: A temporal logic for the interaction of strategies. In: Katoen, J.-P., König, B. (eds.) Proceedings of the 22nd International Conference on Concurrency Theory (CONCUR'11), volume 6901 of Lecture Notes in Computer Science, pp. 466–481. Springer (2011)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.