

On the Boundary of Behavioral Strategies

Fabio Mogavero, Aniello Murano, and Luigi Sauro

Università degli Studi di Napoli Federico II, Napoli, Italy.

fm@fabiomogavero.com murano@na.infn.it luigi.sauro74@gmail.com

Abstract—In the setting of multi-agent games, considerable effort has been devoted to the definition of modal logics for strategic reasoning. In this area, a recent contribution is given by the introduction of *Strategy Logic* (SL, for short) by Mogavero, Murano, and Vardi. This logic allows to reason explicitly about strategies as first order objects and express in a very natural and elegant way several solution concepts like Nash, resilient, and secure equilibria, dominant strategies, etc. The price that one has to pay for the high expressiveness of SL semantics is that agents strategies it admits may be not *behavioral*, i.e., a choice of an agent, at a given moment of a play, may depend on the choices another agent can make in another counterfactual play. As the latter moves are unpredictable, this kind of strategies cannot be synthesized in practice.

In this paper, we investigate two syntactical fragments of SL, namely the *conjunctive-goal* and *disjunctive-goal*, called $SL_{[CG]}$ and $SL_{[DG]}$ for short, and prove that their semantics admit behavioral strategies only. These logics are obtained by forcing SL formulas to be only of the form of conjunctions or disjunctions of goals, which are temporal assertions associated with a binding of agents with strategies. As SL formulas with any Boolean combination of goals turn out to be non behavioral, we have that $SL_{[CG]}$ and $SL_{[DG]}$ represent the maximal fragments of SL describing agent behaviors that are synthesizable. As a consequence of the above results, the model-checking problem for both $SL_{[CG]}$ and $SL_{[DG]}$ is shown to be solvable in 2EXPTIME, as it is for the subsumed logic ATL^* .

I. INTRODUCTION

In recent years, game theory has exhibited to be a fruitful metaphor in open-system verification, where the evolution emerges from the coordination of different parts viewed as autonomous and proactive agents [8], [16]. This has encouraged the development of several modal logics aimed at reasoning about strategies and their interaction [3], [11], [13], [14], [21].

An important contribution in this field has been the development of *Alternating-Time Temporal Logic* (ATL^* , for short), introduced by Alur, Henzinger, and Kupferman [3]. Formally, it is obtained as a generalization of branching-time logic CTL^* [9], where the path quantifiers *there exists* “E” and *for all* “A” are replaced with strategic modalities of the form “ $\langle\langle A \rangle\rangle$ ” and “ $[[A]]$ ”, for a set A of *agents*. These modalities are used to express cooperation and competition among agents in order to achieve a temporal goal. Several decision problems have been investigated about ATL^* . In particular, the model-checking problem is proved to be 2EXPTIME-COMPLETE [3].

Despite its powerful expressiveness, ATL^* suffers from two strong limitations: 1) strategies are treated implicitly through

modalities that refer to games between competing coalitions and 2) strategic modalities only represent coupled $\exists\forall$ and $\forall\exists$ quantifications over strategies. These limitations make ATL^* less suited to formalize several important strategic notions, such as *Nash Equilibrium* and the like.

The above considerations led to introduce and study in [20] a more powerful logic for strategic reasoning in order to “unpack” and extend the ATL^* modalities. The result is *Strategy Logic* (SL, for short), which extends ATL^* in two fundamental aspects. First, strategies are first-order objects that can be existentially and universally quantified. Specifically, SL uses the existential $\langle\langle x \rangle\rangle$ and the universal $[[x]]$ strategic modalities, which can be read as “*there exists a strategy x*” and “*for all strategies x*”, respectively. Second, strategies represent general conditional plans that at each step prescribe an action on the base of the previous history. Thus, strategies are not intrinsically glued to a specific agent, but an explicit *binding* operator (a, x) allows to bind an agent *a* to the strategy associated with a variable *x*.

The SL features allow a finer-grained description of extensive games and reveals aspects that other classic subsumed logics, such as ATL^* , are not able to grasp. In particular, a key aspect in the theory of extensive games is that strategies are *behavioral*, that is, by repeating Myerson’s words: “the correlation between the move he [i.e., an agent] chooses at one information state and the move would have chosen at any other information state has no significance” [22].

However, the expressive power of SL allows to specify some sentences that can be satisfied only by agent strategies that are *not* behavioral. More specifically, in a determined history of the play, what Myerson calls “information state”, the value of an existential quantified strategy may depend on how the other strategies will behave in the future or in other counterfactual plays. This means that, to choose an existential strategy, we need to know the entire structure of universal strategies on all possible histories, which is in general unpredictable, as what we actually know is their behavior on the history of interest only. This means that non-behavioral strategies cannot be synthesized in practice.

Additionally, by maintaining in SL this kind of strategies, we lose important model-theoretic properties and incur in an increased complexity of related decision problems. In particular, in [18], it has been shown that the model-checking problem becomes non-elementary complete. To gain back elementariness, several fragments of SL, strictly subsuming

ATL*, have been investigated and studied in [18], [19]. Among the others, *One-Goal Strategy Logic* (SL[1G], for short) encompasses formulas in a special prenex normal form having a single temporal goal at a time. For a goal, it is specifically meant an SL formula of the type $\mathfrak{b}\psi$, where \mathfrak{b} is a binding prefix of the form $(\alpha_1, x_1), \dots, (\alpha_n, x_n)$ containing all the involved agents and ψ is a linear-time formula. In SL[1G], each temporal formula ψ is prefixed by a quantification-binding prefix $\wp\mathfrak{b}$ that quantifies over a tuple of strategies and binds them to all agents. In [18], [19], it has been showed that SL[1G] admits behavioral strategies¹ only and this has been a key aspect in showing that the model-checking problem for this logic is 2EXPTIME-COMPLETE, as it is for ATL*. This adds motivations on the importance to restrict our attention to behavioral strategies in SL.

In this paper, we achieve the target of identifying sufficient criteria for what is behavioral in SL. Specifically, we consider two syntactical fragments of SL, strictly subsuming SL[1G], namely the *conjunctive-goal* and *disjunctive-goal*, respectively called SL[CG] and SL[DG], for short. These logics are obtained by forcing SL formulas to be only of the form of conjunctions or disjunctions of goals. As main result, we show that also SL[CG] and SL[DG] admit behavioral strategies only. Moreover, since SL formulas with any Boolean combination of goals, named SL[BG] in [18], [19], turn out to be non behavioral, we have that SL[CG] and SL[DG] represent the maximal syntactic fragments of SL that are behavioral. We also solve the model-checking problem for SL[CG] and SL[DG] via alternating tree automata. As for SL[1G], the fact that these fragments are behavioral is a key aspect to strongly simplify the reasoning about strategies by reducing it to a step-by-step reasoning about which action to perform. More precisely, we avoid the projection operation for each quantification in the formula (as it is instead required for SL) by using a dedicated automaton that makes an action quantification for each node of the tree model. As a formula may require different goals to take care of simultaneously, a specific duty of the automaton is to show that such goals can be treated separately, but maintaining their mutual coherence. As this automaton is only exponential in the size of the formula (in particular, it is independent from the alternation number of quantifications in the formula) and its non-emptiness can be computed in exponential time, we get that the model-checking for both SL[CG] and SL[DG] is solvable in 2EXPTIME. It is interesting to observe that the same result holds for SL[1G], while for SL[BG] it still open the question whether the model checking problem is elementary or not.

Besides the theoretical aspects, SL[CG] and SL[DG] enable to formalize several interesting game properties that cannot be expressed in SL[1G], and hence in ATL*. Specifically, on one hand, in SL[CG] it is possible to formalize scenarios

where an agent can join two or more different coalitions without producing mutual conflicts or, somewhat conversely, it can assure that none of them prevails over the others. As it will be clear in the following, in such a case, conjunctive goals are strictly required. As shown in [4], this feature is essential also to address interesting issues such as coalition decomposability, that is whether a whole coalition can be split in smaller, and hence more manageable, sub-coalitions. On the other hand, in SL[DG] it is possible to formalize implication between goals, which can be used to express that a winning condition is weaker than another one.

Due to space limit, most of the proofs are omitted and reported in an extended version. We refer to [18], [19], [20] for more motivations, examples and related material. For other recent works in the field of strategic reasoning, one can also see [1], [2], [5], [6], [7], [10], [17], [23], [25].

Outline: The remaining part of the work is structured as follows. In Section II, we first recall *concurrent game structures*, which represent the semantic framework we use for the addressed logics. Then, we present syntax and semantics of SL and introduce its new fragments SL[CG] and SL[DG], along with some inspiring examples. In addition, we define the notion of *dependence map*, which is used to define the crucial concept of behavioral semantics. In Section III, we show that classical and behavioral semantics for both SL[CG] and SL[DG] coincides. In Section IV, we describe the model-checking automata-theoretic procedure for the introduced fragments and show a 2EXPTIME upper bound for the addressed problem. Finally, at the end of the work, we report some concluding observations and discussions.

II. STRATEGY LOGIC

In this section, we first describe concurrent game structures and give some preliminary notions. Then, we recall syntax and semantics of Strategy Logic (SL, for short) and introduce the new syntactic fragments SL[CG] and SL[DG]. Finally, we define the notions of dependence maps and behavioral semantics. For a detailed introduction on SL, see [18], [20].

A. Concurrent game structures

As in ATL*, SL is interpreted over *concurrent game structures* [3], which are *labeled transition system*, where each state represents a configuration of an extensive game characterized by a set of atomic propositions denoting its meaning. Transitions between states represent possible concurrent moves of the players involved in the game.

Definition II.1 (Concurrent Game Structures). A concurrent game structure (CGS, for short) is a tuple $\mathcal{G} \triangleq \langle \text{AP}, \text{Ag}, \text{Ac}, \text{St}, \lambda, \tau, s_0 \rangle$, where AP and Ag are finite non-empty sets of atomic propositions and agents, Ac and St are enumerable non-empty sets of actions and states, $s_0 \in \text{St}$ is a designated initial state, and $\lambda : \text{St} \rightarrow 2^{\text{AP}}$ is a labeling function that maps each state to the set of atomic propositions true in that state. Let $\text{Dc} \triangleq \text{Ac}^{\text{Ag}}$ be the set of decisions, i.e., functions

¹In [18], [19], we improperly call this kind of strategies *elementary*.

from Ag to Ac representing the choices of an action for each agent.² Then, $\tau : \text{St} \times \text{Dc} \rightarrow \text{St}$ is a transition function mapping a pair of a state and a decision to a state.

In the following, we use the name of a CGS as a subscript to extract the components from its tuple-structure. Accordingly, if $\mathcal{G} = \langle \text{AP}, \text{Ag}, \text{Ac}, \text{St}, \lambda, \tau, s_0 \rangle$, we have that $\text{Ac}_{\mathcal{G}} = \text{Ac}$, $\lambda_{\mathcal{G}} = \lambda$, $s_{0\mathcal{G}} = s_0$, and so on. Before introducing our logic, we need some preliminary definitions.

A *track* (resp., *path*) is a finite (resp., an infinite) sequence of states $\rho \in \text{St}^*$ (resp., $\pi \in \text{St}^\omega$) such that, for all $0 \leq i \leq |\rho| - 1$ (resp., $i \in \mathbb{N}$), there exists a decision $d \in \text{Dc}$ such that $(\rho)_{i+1} = \tau((\rho)_i, d)$ (resp., $(\pi)_{i+1} = \tau((\pi)_i, d)$). The set $\text{Trk} \subseteq \text{St}^+$ (resp., $\text{Pth} \subseteq \text{St}^\omega$) contains all non-zero length tracks (resp., paths). Moreover, $\text{Trk}(s)$ (resp., $\text{Pth}(s)$) indicates the subsets of tracks (resp., paths) starting at a state $s \in \text{St}$. Intuitively, tracks and paths of a CGS \mathcal{G} are legal sequences of reachable states in \mathcal{G} that can be respectively seen as partial and complete history of the game.

A *strategy* is a partial function $f : \text{Trk} \rightarrow \text{Ac}$ that maps each non-zero length track in its domain to an action. Intuitively, a strategy is a conditional plan that for each partial history of \mathcal{G} prescribes an action to be executed. For a state $s \in \text{St}$, a strategy f is said *s-total* if it is defined on all tracks starting in s , i.e., $\text{dom}(f) = \text{Trk}(s)$. The set $\text{Str} \triangleq \text{Trk} \rightarrow \text{Ac}$ (resp., $\text{Str}(s) \triangleq \text{Trk}(s) \rightarrow \text{Ac}$) contains all (resp., *s-total*) strategies.

Let $f \in \text{Str}$ be a strategy and ρ a track in its domain. Then, $(f)_\rho \in \text{Str}$ denotes the *translation* of a strategy f along ρ , i.e. a strategy such that $(f)_\rho(\rho') \triangleq f(\rho \cdot \rho'_{\geq 1})$, if $(\rho')_0 = \text{lst}(\rho)$ and $\rho \cdot \rho'_{\geq 1} \in \text{dom}(f)$, undefined otherwise.³ Intuitively, the translation $(f)_\rho$ is the update of the strategy f , once the history of the game becomes ρ . It is worth noting that, if f is a $(\rho)_0$ -total strategy then $(f)_\rho$ is $\text{lst}(\rho)$ -total.

Let Var be a fixed set of variables. An *assignment* is a partial function $\chi : \text{Var} \cup \text{Ag} \rightarrow \text{Str}$ mapping variables and agents in its domain to a strategy. An assignment χ is *complete* if it is defined on all agents, i.e., $\text{Ag} \subseteq \text{dom}(\chi)$. For a state $s \in \text{St}$, it is said that χ is *s-total* if all strategies $\chi(l)$ are *s-total*, for $l \in \text{dom}(\chi)$. The set $\text{Asg} \triangleq \text{Var} \cup \text{Ag} \rightarrow \text{Str}$ (resp., $\text{Asg}(s) \triangleq \text{Var} \cup \text{Ag} \rightarrow \text{Str}(s)$) contains all (resp., *s-total*) assignments. Moreover, $\text{Asg}(X) \triangleq X \rightarrow \text{Str}$ (resp., $\text{Asg}(X, s) \triangleq X \rightarrow \text{Str}(s)$) indicates the subset of *X-defined* (resp., *s-total*) assignments, i.e., (resp., *s-total*) assignments defined on $X \subseteq \text{Var} \cup \text{Ag}$.

As in the case of strategies, we also define a *translation* along a given track for assignments. For a given state $s \in \text{St}$, let $\chi \in \text{Asg}(s)$ be an *s-total* assignment and $\rho \in \text{Trk}(s)$ a track. Then, $(\chi)_\rho \in \text{Asg}(\text{lst}(\rho))$ denotes the *translation* of χ

²In the following, we use both $X \rightarrow Y$ and Y^X to denote the set of functions from the domain X to the codomain Y .

³By $\text{lst}(w) \triangleq (w)_{|w|-1}$ it is denoted the *last element* of a finite non-empty sequence $w \in \Sigma^+$. Moreover, the notations $(w)_{\leq i} \in \Sigma^*$ and $(w)_{> i} \in \Sigma^\infty$ indicate the *prefix* up to and the *suffix* from index $i \in [0, |w|]$ of a non-empty sequence $w \in \Sigma^\infty$.

along ρ , i.e., the $\text{lst}(\rho)$ -total assignment, with $\text{dom}((\chi)_\rho) \triangleq \text{dom}(\chi)$, such that $(\chi)_\rho(l) \triangleq (\chi(l))_\rho$, for all $l \in \text{dom}(\chi)$.

As in first order logic, in order to quantify over strategies or bind a strategy to an agent, we update an assignment χ by associating an agent or a variable l with a new strategy f . Let $\chi \in \text{Asg}$ be an assignment, $f \in \text{Str}$ a strategy and $l \in \text{Var} \cup \text{Ag}$ either an agent or a variable. Then, $\chi[l \mapsto f] \in \text{Asg}$ denotes the new assignment defined on $\text{dom}(\chi[l \mapsto f]) \triangleq \text{dom}(\chi) \cup \{l\}$ that returns f on l and the same value that χ would return on the rest of its domain. It is worth noting that if χ and f are *s-total* then $\chi[l \mapsto f]$ is also *s-total*.

A *play* is the unique outcome of the game determined by all agent strategies participating to it; formally, given a state $s \in \text{St}$ and a complete *s-total* assignment $\chi \in \text{Asg}(s)$, the function $\text{play}(\chi, s)$ returns the path $\pi \in \text{Pth}(s)$ such that, for all $i \in \mathbb{N}$, it holds that $(\pi)_{i+1} = \tau((\pi)_i, d)$, where $d(a) \triangleq \chi(a)((\pi)_{\leq i})$ for each $a \in \text{Ag}$.

Finally, for a state $s \in \text{St}$ and a complete *s-total* assignment $\chi \in \text{Asg}(s)$, we define the *i-th global translation* that calculates, at a certain step $i \in \mathbb{N}$ of the play, what is the current state and its updated assignment. Formally, the *i-th global translation* of (χ, s) is the pair of a complete assignment and a state $(\chi, s)^i \triangleq ((\chi)_{(\pi)_{\leq i}}, (\pi)_i)$, where $\pi = \text{play}(\chi, s)$.

As in the case of components of a CGS, in order to avoid any ambiguity, we sometimes use the name of the CGS as a subscript of the sets and functions introduced above.

B. Syntax and semantics

Strategy Logic (SL, for short) [20] syntactically extends LTL with two *strategy quantifiers*, the existential $\langle\langle x \rangle\rangle$ and the universal $\llbracket x \rrbracket$, and an *agent binding* (a, x) , where a is an agent and x a variable. Intuitively, these new elements can be respectively read as “*there exists a strategy x*”, “*for all strategies x*”, and “*bind agent a to the strategy associated with the variable x*”.

Definition II.2 (SL Syntax). *Given the set of atomic propositions AP, variables Var, and agents Ag, the formal syntax of SL is defined as follows:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathbf{X}\varphi \mid \varphi \mathbf{U}\varphi \mid \varphi \mathbf{R}\varphi \mid \langle\langle x \rangle\rangle\varphi \mid \llbracket x \rrbracket\varphi \mid (a, x)\varphi;$$

where $p \in \text{AP}$, $x \in \text{Var}$, and $a \in \text{Ag}$.

By $\text{sub}(\varphi)$ we denote the set of subformulas of the formula φ . For instance, the subformulas of $\varphi = \langle\langle x \rangle\rangle(a, x)(\mathbf{F} p)$ are $\text{sub}(\varphi) = \{\varphi, (a, x)(\mathbf{F} p), (\mathbf{F} p), p\}$.

Usually, predicative logics need the concepts of free and bound *placeholders* in order to formally define their semantics. For example, in first order logic the only type of placeholders are variables. In SL, since strategies can be associated to both agents and variables, we use the set of *free agents/variables* $\text{free}(\varphi)$ as the subset of $\text{Ag} \cup \text{Var}$ containing (i) all agents a for which there is no binding (a, x) before the occurrence of a temporal operator and (ii) all variables x for

which there is a binding (a, x) but no quantification $\langle\langle x \rangle\rangle$ or $\llbracket x \rrbracket$. A formula φ without free agents (resp., variables), i.e., with $\text{free}(\varphi) \cap \text{Ag} = \emptyset$ (resp., $\text{free}(\varphi) \cap \text{Var} = \emptyset$), is named *agent-closed* (resp., *variable-closed*). If φ is both agent- and variable-closed, it is named *sentence*. By $\text{snt}(\varphi)$ we denote the set of all sentences that are subformulas of φ .

The SL semantics is defined as follows.

Definition II.3 (SL Semantics). *Given a CGS \mathcal{G} , for all SL formulas φ , states $s \in \text{St}$, and s -total assignments $\chi \in \text{Asg}(s)$ with $\text{free}(\varphi) \subseteq \text{dom}(\chi)$, the modeling relation $\mathcal{G}, \chi, s \models \varphi$ is inductively defined as follows.*

- 1) $\mathcal{G}, \chi, s \models p$ if $p \in \lambda(s)$, with $p \in \text{AP}$.
- 2) Boolean operators are interpreted as usual.
- 3) For a variable $x \in \text{Var}$ and a formula φ , it holds that:
 - a) $\mathcal{G}, \chi, s \models \langle\langle x \rangle\rangle \varphi$ if there exists an s -total strategy $f \in \text{Str}(s)$ such that $\mathcal{G}, \chi[x \mapsto f], s \models \varphi$;
 - b) $\mathcal{G}, \chi, s \models \llbracket x \rrbracket \varphi$ if for all s -total strategies $f \in \text{Str}(s)$ it holds that $\mathcal{G}, \chi[x \mapsto f], s \models \varphi$.
- 4) For an agent $a \in \text{Ag}$, a variable $x \in \text{Var}$, and a formula φ , it holds that $\mathcal{G}, \chi, s \models (a, x) \varphi$ if $\mathcal{G}, \chi[a \mapsto \chi(x)], s \models \varphi$.
- 5) Finally, if the assignment χ is also complete, for all formulas φ, φ_1 , and φ_2 , it holds that:
 - a) $\mathcal{G}, \chi, s \models X \varphi$ if $\mathcal{G}, (\chi, s)^1 \models \varphi$;
 - b) $\mathcal{G}, \chi, s \models \varphi_1 U \varphi_2$ if there is an index $i \in \mathbb{N}$ with $k \leq i$ such that $\mathcal{G}, (\chi, s)^i \models \varphi_2$ and, for all indexes $j \in \mathbb{N}$ with $k \leq j < i$, it holds that $\mathcal{G}, (\chi, s)^j \models \varphi_1$;
 - c) $\mathcal{G}, \chi, s \models \varphi_1 R \varphi_2$ if, for all indexes $i \in \mathbb{N}$ with $k \leq i$, it holds that $\mathcal{G}, (\chi, s)^i \models \varphi_2$ or there is an index $j \in \mathbb{N}$ with $k \leq j < i$ such that $\mathcal{G}, (\chi, s)^j \models \varphi_1$.

It is evident that, due to Items 1, 2, and 3, the LTL semantics is simply embedded into the SL one. Furthermore, since the satisfaction of a sentence φ does not depend on assignments, we omit them and write $\mathcal{G}, s \models \varphi$, when s is a generic state in St , and $\mathcal{G} \models \varphi$ when $s = s_0$.

In what follows a *quantification prefix* over a set $V \subseteq \text{Var}$ of variables is a finite word $\wp \in \{\langle\langle x \rangle\rangle, \llbracket x \rrbracket : x \in V\}^{|\text{V}|}$ of length $|\text{V}|$ such that each variable $x \in V$ occurs just once in \wp . By $\langle\langle \wp \rangle\rangle$ (resp. $\llbracket \wp \rrbracket$) we denote the set of variables occurring *existentially* (resp. *universally*) quantified in \wp . A *binding prefix* b over V is a word of type $(a_1, x_1) \dots (a_m, x_m)$ such that $x_i \in V$, for each $1 \leq i \leq m$, and $\text{Ag} = \{a_1, \dots, a_m\}$. By $\text{Qnt}(V)$ and $\text{Bnd}(V)$ we indicate the set of quantification and binding prefixes over V .

We are now ready to introduce the two fragments of SL we are interested in this paper, i.e., the *conjunctive-goal* and *disjunctive-goal strategy logic* (SL[CG] and SL[DG], for short). In the following, when we have to refer to one of the two logics indifferently, we use the single acronym SL[XG].

Definition II.4 (SL[XG] Syntax). *The syntax of SL[CG] and SL[DG] is defined as follows, with the symbol \otimes used in place of \wedge or \vee , respectively:*

$$\begin{aligned} \varphi ::= p \mid \neg \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid X \varphi \mid \varphi U \varphi \mid \varphi R \varphi \mid \wp \psi \\ \psi ::= b \varphi \mid \psi \otimes \psi. \end{aligned}$$

where $\wp \in \text{Qnt}(\text{free}(\psi))$ and $b \in \text{Bnd}(V)$, for a given set of variables $V \subseteq \text{Var}$.

The name of these fragments comes from the fact that formulas of the kind $b \varphi$ are called goals. It is also evident that SL[XG] resides between the logics SL[1G] (see [19], for more), where ψ consists of a unique goal $b \varphi$, and SL[BG] (see [18], for more), which allows any Boolean combination of goals.

A first important result about SL[XG], is that, differently from SL[1G], it is not invariant under decision-unwinding. Intuitively, by decision-unwinding [19] we mean the tree obtained by unraveling a given CGS with respect to all possible agent decisions (see Appendix B). Consequently, SL[XG] is strictly more expressive than SL[1G].

Theorem II.1 (Unwinding Variance). *There exists an SL[XG] sentence φ and two models \mathcal{G}_1 and \mathcal{G}_2 with isomorphic decision-unwindings such that $\mathcal{G}_1 \models \varphi$ and $\mathcal{G}_2 \not\models \varphi$.*

C. Inspiring examples

We now give a few specification examples, in order to show the expressive power of the introduced fragments.

As first example, assume that one wants to check whether an agent-provider can serve with a unique policy the requests of two agent-clients and the behavior of one agent-client cannot influence the request of the other one. This property can be described by the SL[CG] sentence $\phi = \langle\langle x \rangle\rangle \langle\langle y_2 \rangle\rangle \langle\langle y_3 \rangle\rangle \llbracket z \rrbracket ((a_1, x)(a_2, y_2)(a_3, z) F p \wedge (a_1, x)(a_2, z)(a_3, y_3) F q)$, where a_1 is the provider, a_2 and a_3 are the two clients, and $F p$ and $F q$ are the relative requests. Note that the ATL* formula $\phi_1 = \langle\langle \{a_1, a_2\} \rangle\rangle F p \wedge \langle\langle \{a_1, a_3\} \rangle\rangle F q$ is too weak for this aim, since according to it the strategies that a_1 may need to adopt in the two coalitions could be mutually inconsistent. Similarly, $\phi_2 = \langle\langle \{a_1, a_2, a_3\} \rangle\rangle (F p \wedge F q)$ does not assure that the agent-clients are independent, in the sense that no one of them can jeopardize the request of the other agent. In particular, we have that ϕ implies $\phi_1 \wedge \phi_2$, but the converse is not necessarily true. More generally, sentences like ϕ can be used in several contexts. In the field of multi-agent systems, they tell us that the agent a_1 has a strategy that enables to form two independent coalitions with a_2 and a_3 , respectively, without embarking on a unique coalition formation process with both of them. In the context of formal verification, instead, such sentences can be used at different levels of analysis to test the fault-tolerance of a component with respect to the others or the robustness of an entire system, represented by a_1 , to possible misuses of single users (a_2 and a_3).

Now, consider the CGS \mathcal{G} of Figure 1, where each transition is associated with a triple of binary values representing the decision of a_1 , a_2 , and a_3 , respectively (self-loops have been omitted for simplicity). Let x and

y_2 set to be the strategy that constantly execute the action 1 and y_3 set to be the strategy that constantly execute 0.

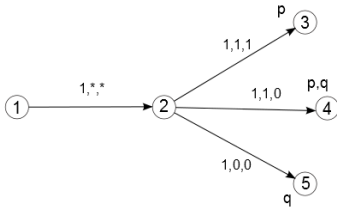


Figure 1. Provider with unique policy.

It is easy to see that, from the initial state s_1 , if x is bound to a_1 and y_2 is bound to a_2 , then the system is forced to reach the states s_3 or s_4 (where p is true). Similarly, if x is bound to a_1 and y_3 is bound to a_3 , the system is forced to reach the states s_4 or s_5 (where q is true). Thus, the previous sentence is satisfied in s_0 .

As another example, let us consider a game with three agents a_1 , a_2 , and a_3 such that a_2 and a_3 want to achieve GFp and GFq , respectively. We want to verify whether a_1 can act as a “super partes” agent that balances the evolution of the game so that the other agents can possibly satisfy their own goal. This can be expressed in $SL[CG]$ by the sentence $\phi' = \langle\langle x \rangle\rangle \llbracket y \rrbracket \langle\langle z_1 \rangle\rangle \langle\langle z_2 \rangle\rangle ((a_1, x)(a_2, y)(a_3, z_1)GFq \wedge (a_1, x)(a_3, y)(a_2, z_2)GFp)$.

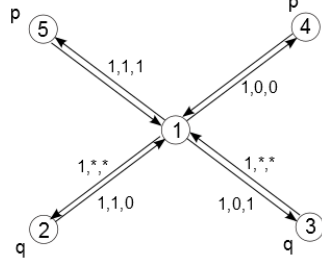


Figure 2. Fair scheduler.

Consider the CGS \mathcal{G} of Figure 2. If a_1 constantly executes the action 1, then the even steps of each possible play will be the initial state s_1 . From that state, a_2 cannot force the next state to yield $\neg q$ and, vice versa, a_3 cannot force the next state to yield $\neg p$. Thus, \mathcal{G} satisfies ϕ' in s_1 . It is worth noting that a_1 does not have the power to decide which winning condition will be satisfied, however by executing the strategy 0^ω it denies both of them.

As final issue, consider the problem to check whether, given a two-player CGS \mathcal{G} , a winning condition ψ_1 is more restrictive for a_1 than another condition ψ_2 . An ATL^* formula such as $\llbracket \{a_1\} \rrbracket \psi_1 \Rightarrow \llbracket \{a_1\} \rrbracket \psi_2$ could be used for this scope. A finer grained representation of the problem consists in analyzing the capability to obtain ψ_1 and ψ_2 for each single strategy of the other agent a_2 : whenever a_1 is able to respond to a strategy of a_2 in order to achieve ψ_1 , then, against that strategy, it can also achieve ψ_2 . Formally, this can be obtained by the $SL[DG]$ sentence $\llbracket x \rrbracket \llbracket y \rrbracket \langle\langle z \rangle\rangle ((a_2, x)(a_1, y)\psi_1 \rightarrow (a_2, x)(a_1, z)\psi_2)$.

D. Strategy quantifications

In this section we formalize the behavioral semantics for the prenex fragment of SL , which plays a key role in maintaining the model-checking problem for $SL[XG]$ as easy as ATL^* . For this scope, we need first to define the concept of *dependence map* that is a Skolemization procedure for SL and, then, the related notion of *elementariness*, which represents a generic functional correspondent of what behavioral means

for strategies. We start with some additional notion. Let $\wp \in \text{Qnt}(V)$ be a quantification prefix over a set $V \subseteq \text{Var}$ of variables. For each $y \in \llbracket \wp \rrbracket$, we use $\text{Dep}(\wp, y)$ to denote the set of universally quantified variables $x \in \llbracket \wp \rrbracket$ that precede y in \wp , that are the variable on which y depends. A *valuation* of variables over a set D is a partial function $v: \text{Var} \rightarrow D$. By $\text{Val}_D(V) \triangleq V \rightarrow D$ we denote the set of all valuation functions over D whose domain is V .

A *dependence map* for \wp over D is a function $\theta: \text{Val}_D(\llbracket \wp \rrbracket) \rightarrow \text{Val}_D(V)$ satisfying the following properties: (i) $\theta(v)(x) = v(x)$, for all $x \in \llbracket \wp \rrbracket$ and (ii), for all $v_1, v_2 \in \text{Val}_D(\llbracket \wp \rrbracket)$ and $y \in \llbracket \wp \rrbracket$, if $v_1 \upharpoonright_{\text{Dep}(\wp, y)} = v_2 \upharpoonright_{\text{Dep}(\wp, y)}$ then $\theta(v_1)(y) = \theta(v_2)(y)$, where $v \upharpoonright_{\text{Dep}(\wp, y)}$ is the restriction of v to $\text{Dep}(\wp, y)$. By $\text{DM}_D(\wp)$ we denote the set of all dependence maps of \wp over D . Intuitively, Item (i) says that θ takes the same values of its argument w.r.t. the universal variables in \wp and Item (ii) ensures that the value of θ w.r.t. an existential variable y in \wp only depends on variables in $\text{Dep}(\wp, y)$.

A fundamental theorem reported below states that if a formula is satisfiable then it is always possible to find a suitable dependence map returning the existential strategies in response to the universal ones. This procedure, easily proved to be correct by induction on the structure of the formula in [18], can be seen as the equivalent of the *Skolemization* in first order logic. Here we give an intuition about it through an example. Consider the $SL[XG]$ sentence $\varphi = \wp\psi$, where $\wp = \llbracket x \rrbracket \langle\langle y \rangle\rangle$ and $\psi = ((a_1, x)(a_2, y)Gp) \otimes ((a_1, y)(a_2, x)F\neg p)$. In order to satisfy φ , we need to choose for every strategy associated with x a “right” strategy to associate with y such that ψ is true. A dependence map over strategies can be given to maintain the correlation between strategies for y chosen with respect to any possible one given for x .

Theorem II.2 (SL Strategy Quantification). *Let \mathcal{G} be a CGS and $\varphi = \wp\psi$ be an SL sentence, where ψ is agent-closed and $\wp \in \text{Qnt}(\text{free}(\psi))$. Then, $\mathcal{G} \models \varphi$ iff there exists a dependence map $\theta \in \text{DM}_{\text{Str}(s_0)}(\wp)$ such that $\mathcal{G}, \theta(x), s_0 \models \psi$, for all $x \in \text{Asg}(\llbracket \wp \rrbracket, s_0)$.*

The above theorem substantially characterizes SL semantics by means of the concept of dependence map. Such a characterization enables the definition of alternative semantics, based on the choice of a subset of dependence maps that ensures better model properties and easier decision problems. Here, we consider the set of dependence maps that are *elementary*, which allows us to greatly simplify the reasoning about strategy quantifications by reducing them to a set of quantifications over actions, one for each track in their domains. Actually, this is a purely functional concept that allows to identify, from a computational point of view, a more tractable set of dependence maps over generic domain set.

We formally define elementariness, through the concept of *adjoint function*. To intuitively understand this concept, consider again the formula φ given above. As we have pointed out in Theorem II.2, a dependence map associates a

strategy for y at every given strategy for x . By recalling that a strategy is a function from tracks to actions, this means that the behavior of the strategy for y on a certain track ρ , depends on the overall behavior of the strategy for x over all possible tracks. The adjoint function lets us to put all possible tracks as a common factor in the choice of strategies for y . Consequently, it will be enough for y to take into account how x behaves on ρ , that is the history of the current play. Let us now formalize the concept of *adjoint function*. From now on, we denote by $\widehat{g} : Y \rightarrow (X \rightarrow Z)$ the operation of *flipping* of a generic function $g : X \rightarrow (Y \rightarrow Z)$, i.e., the transformation of g by swapping the order of its arguments. Let D, T, U , and V be four sets, and $m : (T \rightarrow D)^U \rightarrow (T \rightarrow D)^V$ and $\widetilde{m} : T \rightarrow (D^U \rightarrow D^V)$ two functions. Then, \widetilde{m} is the adjoint of m if $\widetilde{m}(t)(\widehat{g}(t))(x) = m(g)(x)(t)$, for all $g \in (T \rightarrow D)^U$, $x \in V$, and $t \in T$. Thus, a function m transforming a map of kind $(T \rightarrow D)^U$ into a new map of kind $(T \rightarrow D)^V$ has an adjoint \widetilde{m} if such a transformation can be done pointwisely w.r.t. the set T . Similarly, from an adjoint function it is possible to determine the original function unambiguously. Hence, there is a one to one correspondence between functions admitting an adjoint and the adjoint itself.

The formal meaning of the elementariness of a dependence map over generic functions follows.

Definition II.5 (Elementary Dependence Maps). *Let $\wp \in \text{Qnt}(V)$ be a quantification prefix over a set $V \subseteq \text{Var}$ of variables, D and T two sets, and $\theta \in \text{DM}_{T \rightarrow D}(\wp)$ a dependence map for \wp over $T \rightarrow D$. Then, θ is elementary if it admits an adjoint function. $\text{EDM}_{T \rightarrow D}(\wp)$ denotes the set of all elementary dependence maps for \wp over $T \rightarrow D$.*

At this point, as mentioned above, we introduce a notion of *behavioral satisfiability*, in symbols \models_B , which requires the elementariness of dependence maps over strategies.

Definition II.6 (SL Behavioral Semantics). *Let \mathcal{G} be a CGS and $\varphi = \wp\psi$ an SL sentence, where ψ is agent-closed and $\wp \in \text{Qnt}(\text{free}(\psi))$. Then, $\mathcal{G} \models_B \varphi$ iff there exists a dependence map $\theta \in \text{EDM}_{\text{Str}(s_0)}(\wp)$ such that $\mathcal{G}, \theta(\chi), s_0 \models \psi$, for all $\chi \in \text{Asg}(\llbracket \wp \rrbracket, s_0)$.*

Observe that, differently from the classic semantics, the quantifications in a prefix are not treated individually but as an atomic block. This is due to the necessity of having a strict correlation between the point-wise structure of the quantified strategies.

III. BEHAVIORAL SEMANTICS

In this section, we show that in $\text{SL}[\text{XG}]$ standard and behavioral semantics coincide. As seen in the previous section, the concept of elementariness is essentially a property of the dependence maps that allows to identify a way to satisfy a sentence in a behavioral manner. Looking at a prototypical $\text{SL}[\text{XG}]$ sentence $\wp(b_1\psi_1 \otimes \dots \otimes b_n\psi_n)$, with \otimes being either \wedge or \vee , a dependence map, once applied

to the quantification prefix \wp , determines a set of strategies associated with the relative variables. Nevertheless, since each binding b_i possibly “distributes” those strategies to the agents in a different way, we may have n different complete assignments for φ and, hence, as many evolutions of the game. For instance, in the example of Figure 1, if x, y_2 , and y_3 are interpreted as before (i.e., $1^\omega, 1^\omega, 0^\omega$, respectively) and z is set to the strategy that constantly executes the action 1, then $b_1 = (a_1, x)(a_2, y_2)(a_3, z)$ produces the path $\pi_1 = s_1s_2s_3s_3 \dots$ and $b_2 = (a_1, x)(a_2, z)(a_3, y_3)$ the path $\pi_2 = s_1s_2s_4s_4 \dots$. Thus, the main difficulty w.r.t. the proof of behavioral semantics for $\text{SL}[\text{XG}]$ consists in dealing with all these different evolutions coherently. However, the fact that the goals can only be put either in conjunction or in disjunction ensures that these evolutions of the game “run in parallel” without interfering with each other.

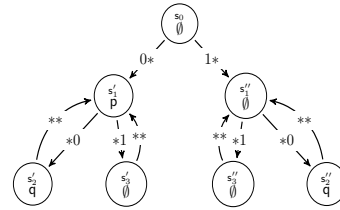


Figure 3. A CGS \mathcal{G} .

On the contrary, the same does not hold for $\text{SL}[\text{BG}]$. Consider, for example, the sentence $\varphi = \llbracket x \rrbracket \langle \langle y \rangle \rangle \langle \langle z \rangle \rangle (a, y)(b, x) \text{Xp} \leftrightarrow (a, z)(b, x) \text{Fq}$ and the CGS \mathcal{G} of Figure 3, where the agent a is the only one that controls the state s_0 , while the agent b controls the states s'_1 and s''_1 . It is not hard to see that $\mathcal{G} \models \varphi$. However, we have that $\mathcal{G} \not\models_B \varphi$, since, to choose the value of the strategies to associate with the existential variables on s_0 , we necessarily need to know the entire strategy associated to x and not only its value on s_0 itself. In particular, agent a should foresight whether agent b intends to achieve, at some point in the future of the states s'_1 and s''_1 , the atomic proposition q or not. Therefore, agent a does not have any synthesizable action in the state s_0 .

As in [18], to prove, instead, that $\text{SL}[\text{XG}]$ admits behavioral strategies only, we reduce the check of $\mathcal{G} \models \varphi$ to a dependence-vs-valuation game, i.e., a turn-based *two player* game (TPG, for short), with the two players called *Even* and *Odd*, which is used to simulate the behavioral semantics. Intuitively, given a CGS \mathcal{G} and an $\text{SL}[\text{XG}]$ formula $\varphi = \wp\psi$, with $\psi = (b_1\psi_1 \otimes \dots \otimes b_n\psi_n)$, a strategy for the player *Even* corresponds to an elementary dependence map, while a strategy for the player *Odd* corresponds to a valuation of the universal variables. Thus, $\mathcal{G} \models_B \varphi$ iff *Even* has a strategy that, no matter how the player *Odd* sets the universal variables, forces the play to satisfy ψ .

A. Dependence-vs-valuation game

Before introducing the formalization of the dependence-vs-valuation game to which we reduce the verification of the $\text{SL}[\text{XG}]$ modeling relation, we need the four auxiliary notions of *walk*, *evolution*, *dependence cluster*, and *valuation cluster* that constitute the building blocks of the game. In

particular, evolutions are used as states of the game arena, while dependence and valuation clusters as possible moves of the two players. In addition, the walks allow to define the winning condition of the game. In the following, \mathcal{G} denotes a CGS, $s \in \text{St}$ a state, and $\wp \in \text{Qnt}(V)$ and $B \subseteq \text{Bnd}(V)$, respectively, a quantification prefix and a set of binding prefixes for a given set of variables $V \subseteq \text{Var}$.

We have emphasized that, when the variables in V have been assigned with some strategies, different bindings produce in general different plays, so we introduce the notion of *walk* to formally deal with these multiple plays at the same time. Given a binding prefix $b \in B$ we denote by $\zeta_b : \text{Ag} \rightarrow V$ the function associating with each agent the related variable in b , i.e., for all $a \in \text{Ag}$, there is $0 \leq i < |b|$ such that $(b)_i = (a, \zeta_b(a))$. Then, a function $\text{wlk} : B \rightarrow \text{Pth}_{\mathcal{G}}$ is a *walk* if there exists an assignment $\chi \in \text{Asg}_{\mathcal{G}}(V, s)$ such that $\text{wlk}(b) = \text{play}_{\mathcal{G}}(\chi \circ \zeta_b, s)$, for all $b \in B$. By $\text{Wlk}_{\mathcal{G}}(B, s)$ we denote the set of all walks of \mathcal{G} from s w.r.t. B . Moreover, by $\text{wlk}_{\mathcal{G}}(\chi, B, s) \in \text{Wlk}_{\mathcal{G}}(B, s)$ we indicate the unique walk derived from χ .

An *evolution* takes a *snapshot* at a certain step i of all tracks that are evolving in a walk, representing in this way a possible position of the verification game behind the check of $\mathcal{G} \models \varphi$. Formally, an evolution is a function $\text{evl} : B \rightarrow \text{St}$ for which there exist a walk $\text{wlk} \in \text{Wlk}_{\mathcal{G}}(B, s)$ and an index $i \in \mathbb{N}$ such that $\text{evl}(b) = (\text{wlk}(b))_i$, for all $b \in B$. By $\text{Evl}(B, s)$ we denote the set of all evolutions of \mathcal{G} from s w.r.t. B .

From now on, let $\text{evl} \in \text{Evl}(B, s)$ be an evolution. Then, a *dependence cluster* w.r.t. evl is a function $\text{dc} : B \rightarrow \text{DM}_{\text{Ac}_{\mathcal{G}}}(\wp)$ such that, for all $b_1, b_2 \in B$, if $\text{evl}(b_1) = \text{evl}(b_2)$ then $\text{dc}(b_1) = \text{dc}(b_2)$. By $\text{DC}_{\mathcal{G}}(\wp, B, \text{evl}, s)$ we denote the set of all dependence clusters for \mathcal{G} in s w.r.t. \wp, B , and evl . Intuitively, a dependence cluster represents, at a certain step, how a dependence map behaves along the tracks produced by different bindings. Since a dependence map produces a unique assignment of variables, if two bindings are going along the same track (which is given by the condition $\text{evl}(b_1) = \text{evl}(b_2)$), the dependence cluster has to return on them the same valuation of existential variables.

Finally, let V' a subset of V . Then, a *valuation cluster* w.r.t. evl and V' is a function $\text{vc} : B \rightarrow \text{Val}_{\text{Ac}_{\mathcal{G}}}(V')$ such that, for all $b_1, b_2 \in B$, if $\text{evl}(b_1) = \text{evl}(b_2)$ then $\text{vc}(b_1) = \text{vc}(b_2)$. By $\text{VC}_{\mathcal{G}}(V', B, \text{evl}, s)$ we denote the set of all valuation clusters for \mathcal{G} in s w.r.t. V', B , and evl . Intuitively, a valuation cluster represents different valuations of universal variables of the quantification prefix along the tracks produced by different bindings. Like dependence clusters, as far as two bindings are running along the same track, valuations of universal variables have to be the same.

At this point, we define the required *dependence-vs-valuation game* that mimics the game behind the behavioral semantics of $\text{SL}[XG]$. Informally, it consists of (i) two non-empty non-intersecting sets of states, N_e and N_o , for players *Even* and *Odd*, respectively, (ii) a designated initial state

$n_0 \in N_e$, and (iii) two binary relations $E_e \subseteq N_e \times N_o$ and $E_o \subseteq N_o \times N_e$ representing at each state the possible moves of Even and Odd, respectively. The notions of track, path, strategy, and play are analogous to those of a CGS and are not repeated here. However, to avoid confusion between concurrent game structures and the corresponding dependence-vs-valuation games, we denote with Mtc the set of all possible paths, namely *matches*, in a dependence-vs-valuation game. Finally, $\text{Win} \subseteq \text{Mtc}$ is the *winning set*, i.e., the set of all paths that are winning for Even.

Definition III.1 (Dependence-vs-Valuation Game). *Let $W \subseteq \text{Wlk}_{\mathcal{G}}(B, s)$ be a set of walks. Then, the dependence-vs-valuation game for \mathcal{G} in s over W w.r.t. \wp and B is the TPG $\mathcal{H}(\mathcal{G}, s, W, \wp, B) \triangleq \langle N_e, N_o, E_e, E_o, n_0, \text{Win} \rangle$ where:*

- $N_e \triangleq \text{Evl}_{\mathcal{G}}(B, s)$ and $N_o \triangleq \{(\text{evl}, \text{dc}) : \text{evl} \in \text{Evl}_{\mathcal{G}}(B, s) \wedge \text{dc} \in \text{DC}_{\mathcal{G}}(\wp, B, \text{evl}, s)\}$;
- $E_e \triangleq \{(\text{evl}, (\text{evl}, \text{dc})) \in N_e \times N_o\}$ and $E_o \triangleq \{((\text{evl}, \text{dc}), \text{evl}') \in N_o \times N_e : \exists \text{vc} \in \text{VC}_{\mathcal{G}}(\llbracket \wp \rrbracket, B, \text{evl}, s) \text{ s.t. } \forall b \in B. \text{evl}'(b) = \tau(\text{evl}(b), \text{dc}(b)(\text{vc}(b)) \circ \zeta_b)\}$;
- $n_0(b) \triangleq s$, for all $b \in B$;
- $\text{Win} \triangleq \{\varpi \in \text{Mtc} : \exists \text{wlk} \in W. \forall i \in \mathbb{N}, b \in B. (\varpi)_{2i}(b) = (\text{wlk}(b))_i\}$.

Intuitively, the previous definition reformulates the verification process of a sentence φ as a zero-sum game where, step-by-step, the Even player attempts to satisfy φ by choosing “good” dependence clusters (Odd nodes) that, for each binding, interpret the existential variables and the Odd player replies with “bad” valuations of the universal variables (Even nodes) to try to make φ unsatisfied. In particular, the peculiarity of behavioral semantics is reformulated in this setting, through the concept of elementary dependence map, by means of the following notion of *encasement*.

Definition III.2 (Encasement). *Let $W \subseteq \text{Wlk}_{\mathcal{G}}(B, s)$ be a set of walks. Then, W is an encasement w.r.t. \wp and B if there exists an elementary dependence map $\theta \in \text{EDM}_{\text{Str}_{\mathcal{G}}(s)}(\wp)$ such that, for all assignments $\chi \in \text{Asg}_{\mathcal{G}}(\llbracket \wp \rrbracket, s)$, it holds that $\text{wlk}_{\mathcal{G}}(\theta(\chi), B, s) \in W$.*

B. Proof of behavioral semantics

In this section, we describe the proof of the behavioral semantics for $\text{SL}[XG]$, which allows us to construct, in the next section, a suitable automata-theoretic procedure for its model-checking problem. Note that, due to the length and high-complex nature of the whole proof, we only sketch its fundamental steps and refer to Appendix C for full details.

From a very high-level point of view, the proof schema proceeds in a similar way to what was done in [18] for $\text{SL}[1G]$. Therefore, in order to simplify the presentation and better understand the new proof, we first recall the structure of the old one: (i) we start constructing a two player game \mathcal{H} simulating $\text{SL}[1G]$ behavioral semantics of a given sentence $\phi = \wp b \psi$ over an assigned CGS \mathcal{G} ; (ii) let P be the set of paths

satisfying the LTL formula ψ , which is also the winning set of the game \mathcal{H} , we prove that it is an encasement iff $\mathcal{G} \models_{\mathcal{B}} \phi$ (see Definition 3.4 of [18]); (iii) via the encasement property, we show that if player Even wins \mathcal{H} then \mathcal{P} is an encasement and, vice versa, if \mathcal{P} is Borelian but not an encasement then player Odd wins \mathcal{H} (encasement characterization lemma, see Lemma B.5 of [18]); (iv) we prove that if player Odd wins the dual game $\overline{\mathcal{H}}$, obtained by the dual prefix $\overline{\varphi}$, then player Even wins the original game \mathcal{H} (dependence-vs-valuation duality lemma, see Lemma B.4 of [18]); (v) finally, we put all these facts together in the following way (see Theorem 4.24 of [18]): (a) we suppose that $\mathcal{G} \models \phi$ and assume $\overline{\mathcal{P}}$ as the set of paths satisfying $\neg\psi$, which is also the Borelian winning set of the dual game $\overline{\mathcal{H}}$; (b) since $\mathcal{G} \not\models_{\mathcal{B}} \neg\phi$, by Item (ii), we get that $\overline{\mathcal{P}}$ is not an encasement; (c) thus, by Item (iii), we derive that player Odd wins $\overline{\mathcal{H}}$; (d) so, by Item (iv), we have that player Even wins the game \mathcal{H} ; (e) now, again by Item (iii), we get that \mathcal{P} is an encasement; (f) finally, by a last application of Item (ii), we conclude that $\mathcal{G} \models_{\mathcal{B}} \phi$. At the end, since the inverse direction, $\mathcal{G} \models_{\mathcal{B}} \phi$ implies $\mathcal{G} \models \phi$, is immediate by definition, we get that $\mathcal{G} \models \phi$ iff $\mathcal{G} \models_{\mathcal{B}} \phi$.

Clearly, due to the fact that an $\text{SL}[\text{XG}]$ sentence has to deal with more than one play at a time, in order to apply the same proof structure to this logic, we have to generalize the two player game described above. Then, walks and evolutions allow to coherently track the verification process of temporal operators on all plays derived by the bindings, in one shot.

However, it is important to observe that parameterizing all two-player game components w.r.t. bindings does not entail by itself that the proof for $\text{SL}[\text{IG}]$ works directly for $\text{SL}[\text{XG}]$. Indeed, another fundamental aspects in our new approach is to show that when one of the two players has a winning strategy, he has a decoupled winning strategy, i.e., a way to chose his moves w.r.t. a given binding that does not depend on what the other player has done before on bindings that are not running on the same track. To formally prove this fact, we first introduce a *topology of open and closed* sets on walks and then prove the following two preliminary facts: (i) the set of walks derived by a $\text{SL}[\text{XG}]$ sentence is a member of the topology; (ii) if a player wins the two player game with an open/closed winning set then he has a decoupled winning strategy. Informally, a set of walks W is open if, for each walk $\text{wlk} \in W$, there exists a binding b for which the play $\text{wlk}(b)$, whatever it is combined with other plays for the other bindings, form a new walk still contained in W . On the contrary, a set of walks W is closed if every walk wlk obtained by the shuffle of the plays of a set of walks $\{\text{wlk}_1, \dots, \text{wlk}_n\}$ in W is still a member of W . With this results as a tool, we are able to show the following generalizations of Lemmas B.5 and B.4 of [18], respectively.

Lemma III.1 (Encasement Characterization). *Let $W \subseteq \text{Wlk}_{\mathcal{G}}(\mathcal{B}, s)$ be an open/closed set of walks and $\mathcal{H} = \langle \mathcal{A}, \text{Win} \rangle = \mathcal{H}(\mathcal{G}, s, W, \varphi, \mathcal{B})$ be the dependence-vs-valuation game for*

\mathcal{G} in s over W w.r.t. φ and \mathcal{B} . Then, the following hold:

- 1) *if player Even wins \mathcal{H} then W is an encasement w.r.t. φ and \mathcal{B} ;*
- 2) *if Win is a Borelian set and W is not an encasement w.r.t. φ and \mathcal{B} then player Odd wins \mathcal{H} .*

Lemma III.2 (Dependence-vs-Valuation Duality). *Let $W \subseteq \text{Wlk}_{\mathcal{G}}(\mathcal{B}, s)$ be an open/closed set of walks, $\mathcal{H} = \langle \mathcal{A}, \text{Win} \rangle = \mathcal{H}(\mathcal{G}, s, W, \varphi, \mathcal{B})$ the dependence-vs-valuation game for \mathcal{G} in s over W w.r.t. φ and \mathcal{B} , and $\overline{\mathcal{H}} = \langle \overline{\mathcal{A}}, \text{Mtc} \setminus \text{Win} \rangle = \mathcal{H}(\mathcal{G}, s, \text{Wlk}_{\mathcal{G}}(\mathcal{B}, s) \setminus W, \overline{\varphi}, \mathcal{B})$ its dual game. Then, if player Odd wins the dual TPG $\overline{\mathcal{H}}$, player Even wins the TPG \mathcal{H} .*

At this point, we can sketch the proof of behavioral semantics for $\text{SL}[\text{CG}]$ (resp., $\text{SL}[\text{DG}]$): (a) we suppose that $\mathcal{G} \models \phi$, where $\phi = \varphi \bigwedge_{b \in \mathcal{B}} b\psi_b$ (resp., $\phi = \varphi \bigvee_{b \in \mathcal{B}} b\psi_b$) and assume \overline{W} as the set of walks satisfying $\neg \bigwedge_{b \in \mathcal{B}} b\psi_b$ (resp., $\neg \bigvee_{b \in \mathcal{B}} b\psi_b$), which is linked one-to-one to the Borelian winning set $\text{Mtc} \setminus \text{Win}$ of the dual game $\overline{\mathcal{H}}$; (b) we show that \overline{W} is an open (resp., closed) set, so, $W \triangleq \text{Wlk}_{\mathcal{G}}(\mathcal{B}, s) \setminus \overline{W}$ is closed (resp., open); (c) since $\mathcal{G} \not\models_{\mathcal{B}} \neg\phi$, we obtain that $\text{Mtc} \setminus \text{Win}$ is not an encasement; (d) therefore, by Item 2 of Lemma III.1, we derive that player Odd wins $\overline{\mathcal{H}}$; (e) consequently, by Lemma III.2, we have that player Even wins the original game \mathcal{H} ; (f) now, by Item 1 of Lemma III.1, we obtain that the set of walks W , from which \mathcal{H} is derived, is an encasement; (g) finally we conclude that $\mathcal{G} \models_{\mathcal{B}} \phi$.

Theorem III.1 ($\text{SL}[\text{XG}]$ Behavioral Semantics). *For all $\text{SL}[\text{XG}]$ sentences φ , it holds that $\mathcal{G} \models \varphi$ iff $\mathcal{G} \models_{\mathcal{B}} \varphi$.*

Finally, note that the same two player game we have defined for $\text{SL}[\text{XG}]$ could also be used to prove the behavioral semantics of the full $\text{SL}[\text{BG}]$, as its structure does not change when we have generic Boolean combinations of goals. However, since we already know that $\text{SL}[\text{BG}]$ admits non-behavioral strategies too, an immediate question that promptly arises is why the proof does not work for the latter logic. Essentially, the problem is that the winning set derived from a $\text{SL}[\text{BG}]$ sentence does not satisfy the decoupled property we mentioned above. This is due to the fact that the induced sets of walks are neither open nor closed, but belong to higher levels of the related topological hierarchy. Consequently, Lemmas III.1 and III.2 cannot hold for $\text{SL}[\text{BG}]$.

IV. MODEL-CHECKING PROCEDURE

We finally solve the model-checking problem for $\text{SL}[\text{XG}]$ and show that it is 2EXPTIME-COMPLETE, as it is for the less expressive ATL^* and $\text{SL}[\text{IG}]$ logics. The algorithmic procedure is based on an automata-theoretic approach, which reduces the decision problem for our logic to the emptiness problem of a suitable universal Co-Büchi tree automaton (UCT, for short) [12]. Our technique is innovative w.r.t. those proposed in literature for CTL^* [15] and ATL^* [3], since it is based on the novel notion of elementary dependence map and behavioral semantics. In particular, we extend the procedure

proposed in [18] for SL[1G], along with a machinery to handle the conjunction/disjunction of goals. The high-level idea behind this approach is to avoid the use of projections for the strategy quantifications (which is instead required for SL), by reducing them to action quantifications, which can be managed on each state of the model without a non-elementary blow-up. Naturally, this approach is correct, since SL[XG] has behavioral strategies only, as we proved before.

To proceed with the formal description of the model-checking procedure, we first introduce the concept of encoding for the assignments over a CGS.

Intuitively, this is a tree \mathcal{T}_χ whose nodes correspond to all possible histories in the unraveling of the CGS \mathcal{G} and whose labeling represents, for the given assignment χ , all actions that the strategies associated to the variables prescribe.

Definition IV.1 (Assignment-State Encoding). *Let \mathcal{G} be a CGS, $s \in \text{St}_{\mathcal{G}}$ one of its states, and $\chi \in \text{Asg}_{\mathcal{G}}(\mathbb{V}, s)$ an assignment defined on the set $\mathbb{V} \subseteq \text{Var} \cup \text{Ag}$. Then, a $(\text{Val}_{\text{Acg}}(\mathbb{V}) \times \text{St}_{\mathcal{G}})$ -labeled $\text{St}_{\mathcal{G}}$ -tree $\mathcal{T}_\chi \triangleq \langle \mathbb{T}, \mathbb{u} \rangle$, where $\mathbb{T} \triangleq \{\rho_{\geq 1} : \rho \in \text{Trk}_{\mathcal{G}}(s)\}$, is an assignment-state encoding for χ if it holds that $\mathbb{u}(t) \triangleq (\widehat{\chi}(s \cdot t), \text{lst}(s \cdot t))$, for all $t \in \mathbb{T}$.*

By a suitably embedding of the Vardi-Wolper construction [24] into a tree automaton, we build an UCT recognizing all assignment-state encodings derived by assignments satisfying a given goal.

Lemma IV.1 (Goal Automaton). *Let \mathcal{G} be a CGS and $b\psi$ a goal without principal subsentences. There is a UCT $\mathcal{U}_{b\psi}^{\mathcal{G}}$ with $0(2^{|\psi|})$ states such that, for all states $s \in \text{St}_{\mathcal{G}}$ and assignments $\chi \in \text{Asg}_{\mathcal{G}}(\text{free}(b\psi), s)$, it holds that $\mathcal{G}, \chi, s \models b\psi$ iff $\mathcal{T}_\chi \in \text{L}(\mathcal{U}_{b\psi}^{\mathcal{G}})$, where \mathcal{T}_χ is the assignment-state encoding for χ and $\text{L}(\mathcal{U}_{b\psi}^{\mathcal{G}})$ is the set of trees $\mathcal{U}_{b\psi}^{\mathcal{G}}$ accepts.*

Now, we introduce an encoding for the information contained into the elementary dependence maps over strategies used to satisfy a given sentence. Intuitively, this is a tree similar to the assignment-state encoding, except that the labeling is used to represent the dependence maps over actions contained into the dependence map over strategies.

Definition IV.2 (Elementary Dependence-State Encoding). *Let \mathcal{G} be a CGS, $s \in \text{St}_{\mathcal{G}}$ one of its states, and $\theta \in \text{EDM}_{\text{Str}_{\mathcal{G}}(s)}(\wp)$ an elementary dependence map over strategies for a quantification prefix $\wp \in \text{Qnt}(\mathbb{V})$ over the set $\mathbb{V} \subseteq \text{Var}$. Then, a $(\text{DM}_{\text{Acg}}(\wp) \times \text{St}_{\mathcal{G}})$ -labeled $\text{St}_{\mathcal{G}}$ -tree $\mathcal{T}_\theta \triangleq \langle \mathbb{T}, \mathbb{u} \rangle$, where $\mathbb{T} \triangleq \{\rho_{\geq 1} : \rho \in \text{Trk}_{\mathcal{G}}(s)\}$, is an elementary dependence-state encoding for θ if it holds that $\mathbb{u}(t) \triangleq (\theta(s \cdot t), \text{lst}(s \cdot t))$, for all $t \in \mathbb{T}$.*

In the next lemma, we show the existence of an UCT that accepts a given elementary dependence-state encoding \mathcal{T} iff an input UCT accepts all assignment-state encodings \mathcal{T}' derived from \mathcal{T} . This automaton is used to handle the strategy quantifications on each state of the model, by means

of quantification over actions modeled by the choice of an action dependence map.

Lemma IV.2 (Dependence Map Automaton). *Let \mathcal{G} be a CGS, \mathcal{U} a UCT, and $\wp \in \text{Qnt}(\mathbb{V})$ a quantification prefix over $\mathbb{V} \subseteq \text{Var}$. There is a UCT \mathcal{U}_\wp with the same states of \mathcal{U} such that, for all states $s \in \text{St}_{\mathcal{G}}$ and elementary dependence maps over strategies $\theta \in \text{EDM}_{\text{Str}_{\mathcal{G}}(s)}(\wp)$, it holds that $\mathcal{T}_\theta \in \text{L}(\mathcal{U}_\wp)$ iff $\mathcal{T}_{\theta(\chi)} \in \text{L}(\mathcal{U})$, for all $\chi \in \text{Asg}_{\mathcal{G}}(\llbracket \wp \rrbracket, s)$, where \mathcal{T}_θ and $\mathcal{T}_{\theta(\chi)}$ are, respectively, the elementary dependence-state encoding for θ and the assignment-state encoding for $\theta(\chi)$.*

We can now state the next theorem that is at the base of the model-checking procedure for SL[XG]. Actually, we build an automaton for an SL[CG] sentence and translate the model-checking question to its non-emptiness problem. In the case we start with an SL[DG] sentence, first we dualize it in an SL[CG] one, apply the previous construction, and check for the emptiness of the obtained automaton. Thus, the result for SL[DG] follows merely as a corollary.

Theorem IV.1 (SL[XG] Sentence Automaton). *Let \mathcal{G} be a CGS, $s \in \text{St}_{\mathcal{G}}$ one of its states, and $\phi = \wp \bigwedge_{b \in \mathbb{B}} b\psi_b$ (resp., $\phi = \wp \bigvee_{b \in \mathbb{B}} b\psi_b$) an SL[CG] (resp., SL[DG]) sentence. Then, there exists an UCT $\mathcal{U}_\phi^{\mathcal{G}, s}$ with $0(2^{|\phi|})$ states such that $\mathcal{G}, \emptyset, s \models \phi$ iff $\text{L}(\mathcal{U}_\phi^{\mathcal{G}, s}) \neq \emptyset$ (resp., $\text{L}(\mathcal{U}_\phi^{\mathcal{G}, s}) = \emptyset$).*

Finally, by a simple calculation of the size of \mathcal{U}_ϕ and the complexity of the related (non) emptiness problem, we state in the next theorem the exact complexity of the model-checking problem for SL[XG]. Note that, in order to maintain a low data complexity, we first translate the UCT into an NPT and then make the product with the CGS under analysis.

Theorem IV.2 (SL[XG] Model-Checking). *The model-checking problem for SL[XG] is PTIME-COMplete w.r.t. the size of the model and 2EXPTIME-COMplete w.r.t. the size of the specification.*

V. DISCUSSION

Reasoning implicitly about strategies, as done in ATL* and its several variants, works well for two persons/teams zero-sum games, where at the abstract level what matters are simply the outcomes that players can ensure. However, for multi-player games with non-zero-sum objectives, it has been observed that an explicit representation of strategies can be useful in several cases to better capture the involved reasoning and in expressing various solution concepts, such as Nash, resilient, and secure equilibria. It is for this reason that a first-order framework such as SL, which allows quantification over strategy terms, is a natural setting to explore. Given this context, SL seems somehow even too expressive in the sense that it is possible to formulate sentences that can be satisfied only if we accept non-behavioral existentially quantified strategies. Such sentences can be considered as the *undesirable* part of SL for several reasons. First, their

satisfaction joins together different possible plays of a game, whereas in standard game theory each single play ends up with an utility profile or a winning agent, so they do not have a clear game-theoretical counterpart. Second, their meaning is somewhat deceptive: even when they say that an agent can in principle respond to another agent, actually the relative strategy is not synthesizable, since it requires information that is inherently not at the disposal of that agent. Finally, also from a computational point of view, all relative reasoning tasks are severely affected. In particular, model checking is non-elementary in the size of the specification.

Clearly, all these considerations arise the problem to isolate the *desirable* part of SL that requires behavioral strategies only. This property essentially says that in order to reason about strategies it suffices to reason about individual actions, for each history of the game. Thanks to this, we are able to employ an automata-theoretic approach through which the strategy quantifications can be handled without the need to perform the projection operations, which usually implies a non-elementary blow up of the resulting automaton.

In this paper, we have addressed the question of which are the SL syntactic fragments of this sort. Specifically, we have introduced and investigated the conjunctive-goal and disjunctive-goal SL, respectively called SL[CG] and SL[DG], and shown that they admit a behavioral semantics. Since by allowing any Boolean combination of goals, the resulting logic, named SL[BG], does not retain this property [18], we have that SL[CG] and SL[DG] are the maximal syntactic fragments of SL that are behavioral. Moreover, we proved that their model-checking problem is 2EXPTIME-COMPLETE, i.e., asymptotically not more expensive than the one for SL[1G], which strictly subsumes ATL*. This strengthens the hypothesis that being behavioral is a sufficient condition to maintain the same complexity of ATL*.

Finally, it is important to stress that the fragments we have introduced in this work can be considered as the first relevant attempt to define a logic formalism able to describe synthesizable properties of non-zero-sum games, such as coalition decomposability, arbitrage and implication of winning conditions, etc., still maintaining a 2EXPTIME-COMPLETE model-checking problem.

ACKNOWLEDGMENTS

We thank G. De Giacomo and M.Y. Vardi for useful discussions on the concept of behavioral strategies.

REFERENCES

- [1] T. Agotnes, V. Goranko, and W. Jamroga. Alternating-Time Temporal Logics with Irrevocable Strategies. In *TARK'07*, pages 15–24, 2007.
- [2] T. Agotnes and D. Walther. A Logic of Strategic Ability Under Bounded Memory. *JLLI*, 18(1):55–77, 2009.
- [3] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-Time Temporal Logic. *JACM*, 49(5):672–713, 2002.
- [4] G. Boella, L. Sauro, and L. van der Torre. Strengthening Admissible Coalitions. In *ECAI'06*, pages 195–199. IOS Press, 2006.
- [5] L. Bozzelli, A. Murano, and A. Peron. Pushdown module checking. *Form. Meth. in Sys. Des.*, 36(1):65–95, 2010.
- [6] T. Brihaye, A. Da Costa Lopes, F. Laroussinie, and N. Markey. ATL with Strategy Contexts and Bounded Memory. In *LFCS'09*, LNCS 5407, pages 92–106. Springer, 2009.
- [7] K. Chatterjee, T.A. Henzinger, and N. Piterman. Strategy Logic. *IC*, 208(6):677–693, 2010.
- [8] E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking*. MIT Press, 2002.
- [9] E.A. Emerson and J.Y. Halpern. “Sometimes” and “Not Never” Revisited: On Branching Versus Linear Time. *JACM*, 33(1):151–178, 1986.
- [10] B. Finkbeiner and S. Schewe. Coordination Logic. In *CSL'10*, LNCS 6247, pages 305–319. Springer, 2010.
- [11] S. Ghosh and R. Ramanujam. Strategies in Games: A Logic-Automata Study. In *ESSLLI'11*, LNCS 7388, pages 110–159. Springer, 2011.
- [12] E. Grädel, W. Thomas, and T. Wilke. *Automata, Logics, and Infinite Games: A Guide to Current Research*. LNCS 2500. Springer, 2002.
- [13] W. Jamroga and W. van der Hoek. Agents that Know How to Play. *FI*, 63(2-3):185–219, 2004.
- [14] Wojciech Jamroga and Wojciech Penczek. Specification and Verification of Multi-Agent Systems. In *ESSLLI'11*, LNCS 7388, pages 210–263. Springer, 2011.
- [15] O. Kupferman, M.Y. Vardi, and P. Wolper. An Automata Theoretic Approach to Branching-Time Model Checking. *JACM*, 47(2):312–360, 2000.
- [16] O. Kupferman, M.Y. Vardi, and P. Wolper. Module Checking. *IC*, 164(2):322–344, 2001.
- [17] A. Da Costa Lopes, F. Laroussinie, and N. Markey. ATL with Strategy Contexts: Expressiveness and Model Checking. In *FSTTCS'10*, LIPIcs 8, pages 120–132, 2010.
- [18] F. Mogavero, A. Murano, G. Perelli, and M.Y. Vardi. Reasoning About Strategies: On the Model-Checking Problem. Technical Report 1112.6275, arXiv, December 2011.
- [19] F. Mogavero, A. Murano, G. Perelli, and M.Y. Vardi. What Makes ATL* Decidable? A Decidable Fragment of Strategy Logic. In *CONCUR'12*, LNCS 7454, pages 193–208. Springer, 2012.
- [20] F. Mogavero, A. Murano, and M.Y. Vardi. Reasoning About Strategies. In *FSTTCS'10*, LIPIcs 8, pages 133–144, 2010.
- [21] F. Mogavero, A. Murano, and M.Y. Vardi. Relentful Strategic Reasoning in Alternating-Time Temporal Logic. In *LPAR'10*, LNAI 6355, pages 371–387. Springer, 2010.
- [22] R.B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, 1991.
- [23] S. Pinchinat. A Generic Constructive Solution for Concurrent Games with Expressive Constraints on Strategies. In *ATVA'07*, LNCS 4762, pages 253–267. Springer, 2007.
- [24] M.Y. Vardi and P. Wolper. An Automata-Theoretic Approach to Automatic Program Verification. In *LICS'86*, pages 332–344. IEEE Computer Society, 1986.
- [25] D. Walther, W. van der Hoek, and M. Wooldridge. Alternating-Time Temporal Logic with Explicit Strategies. In *TARK'07*, pages 269–278, 2007.

A. Mathematical Notation

In this short reference appendix, we report the classical mathematical notation and some common definitions that are used along the whole work.

Classic objects: We consider \mathbb{N} as the set of *natural numbers* and $[m, n] \triangleq \{k \in \mathbb{N} : m \leq k \leq n\}$, $[m, n[\triangleq \{k \in \mathbb{N} : m \leq k < n\}$, $]m, n] \triangleq \{k \in \mathbb{N} : m < k \leq n\}$, and $]m, n[\triangleq \{k \in \mathbb{N} : m < k < n\}$ as its *interval subsets*, with $m \in \mathbb{N}$ and $n \in \widehat{\mathbb{N}} \triangleq \mathbb{N} \cup \{\omega\}$, where ω is the *numerable infinity*, i.e., the *least infinite ordinal*. Given a *set X of objects*, we denote by $|X| \in \widehat{\mathbb{N}} \cup \{\infty\}$ the *cardinality* of X , i.e., the number of its elements, where ∞ represents a *more than countable* cardinality, and by $2^X \triangleq \{Y : Y \subseteq X\}$ the *powerset* of X , i.e., the set of all its subsets.

Relations: By $R \subseteq X \times Y$ we denote a *relation* between the *domain* $\text{dom}(R) \triangleq X$ and *codomain* $\text{cod}(R) \triangleq Y$, whose *range* is indicated by $\text{rng}(R) \triangleq \{y \in Y : \exists x \in X. (x, y) \in R\}$. We use $R^{-1} \triangleq \{(y, x) \in Y \times X : (x, y) \in R\}$ to represent the *inverse* of R itself. Moreover, by $S \circ R$, with $R \subseteq X \times Y$ and $S \subseteq Y \times Z$, we denote the *composition* of R with S , i.e., the relation $S \circ R \triangleq \{(x, z) \in X \times Z : \exists y \in Y. (x, y) \in R \wedge (y, z) \in S\}$. We also use $R^n \triangleq R^{n-1} \circ R$, with $n \in [1, \omega[$, to indicate the *n-iteration* of $R \subseteq X \times Y$, where $Y \subseteq X$ and $R^0 \triangleq \{(y, y) : y \in Y\}$ is the *identity* on Y . With $R^+ \triangleq \bigcup_{n=1}^{\omega} R^n$ and $R^* \triangleq R^+ \cup R^0$ we denote, respectively, the *transitive* and *reflexive-transitive closure* of R . Finally, for an *equivalence relation* $R \subseteq X \times X$ on X , we represent with $(X/R) \triangleq \{[x]_R : x \in X\}$, where $[x]_R \triangleq \{x' \in X : (x, x') \in R\}$, the *quotient* set of X w.r.t. R , i.e., the set of all related *equivalence classes* $[\cdot]_R$.

Functions: We use the symbol $Y^X \subseteq 2^{X \times Y}$ to denote the set of *total functions* f from X to Y , i.e., the relations $f \subseteq X \times Y$ such that for all $x \in \text{dom}(f)$ there is exactly one element $y \in \text{cod}(f)$ such that $(x, y) \in f$. Often, we write $f : X \rightarrow Y$ and $f : X \dashrightarrow Y$ to indicate, respectively, $f \in Y^X$ and $f \in \bigcup_{X' \subseteq X} Y^{X'}$. Regarding the latter, note that we consider f as a *partial function* from X to Y , where $\text{dom}(f) \subseteq X$ contains all and only the elements for which f is defined. Given a set Z , by $f|_Z \triangleq f \cap (Z \times Y)$ we denote the *restriction* of f to the set $X \cap Z$, i.e., the function $f|_Z : X \cap Z \rightarrow Y$ such that, for all $x \in \text{dom}(f) \cap Z$, it holds that $f|_Z(x) = f(x)$. Moreover, with \emptyset we indicate a generic *empty function*, i.e., a function with empty domain. Note that $X \cap Z = \emptyset$ implies $f|_Z = \emptyset$. Finally, for two partial functions $f, g : X \dashrightarrow Y$, we use $f \uplus g$ and $f \cap g$ to represent, respectively, the *union* and *intersection* of these functions defined as follows: $\text{dom}(f \uplus g) \triangleq \text{dom}(f) \cup \text{dom}(g) \setminus \{x \in \text{dom}(f) \cap \text{dom}(g) : f(x) \neq g(x)\}$, $\text{dom}(f \cap g) \triangleq \{x \in \text{dom}(f) \cap \text{dom}(g) : f(x) = g(x)\}$, $(f \uplus g)(x) = f(x)$ for $x \in \text{dom}(f \uplus g) \cap \text{dom}(f)$, $(f \uplus g)(x) = g(x)$ for $x \in \text{dom}(f \uplus g) \cap \text{dom}(g)$, and $(f \cap g)(x) = f(x)$ for $x \in \text{dom}(f \cap g)$.

Words: By X^n , with $n \in \mathbb{N}$, we denote the set of all *n-tuples* of elements from X , by $X^* \triangleq \bigcup_{n=0}^{\omega} X^n$ the set of *finite words* on the *alphabet* X , by $X^+ \triangleq X^* \setminus \{\varepsilon\}$ the set of *non-empty words*, and by X^ω the set of *infinite words*, where, as usual, $\varepsilon \in X^*$ is the *empty word*. The *length* of a word $w \in X^\infty \triangleq X^* \cup X^\omega$ is represented with $|w| \in \widehat{\mathbb{N}}$. By $(w)_i$ we indicate the *i-th letter* of the finite word $w \in X^*$, with $i \in [0, |w|[$. Furthermore, by $\text{fst}(w) \triangleq (w)_0$ (resp., $\text{lst}(w) \triangleq (w)_{|w|-1}$), we denote the *first* (resp., *last*) letter of w . In addition, by $(w)_{\leq i}$ (resp., $(w)_{> i}$), we indicate the *prefix* up to (resp., *suffix* after) the letter of index i of w , i.e., the finite word built by the first $i+1$ (resp., last $|w|-i-1$) letters $(w)_0, \dots, (w)_i$ (resp., $(w)_{i+1}, \dots, (w)_{|w|-1}$). We also set, $(w)_{< 0} \triangleq \varepsilon$, $(w)_{< i} \triangleq (w)_{\leq i-1}$, $(w)_{\geq 0} \triangleq w$, and $(w)_{\geq i} \triangleq (w)_{> i-1}$, for $i \in [1, |w|[$. Mutatis mutandis, the notations of *i-th letter*, *first*, *prefix*, and *suffix* apply to infinite words too. Finally, by $\text{pfX}(w_1, w_2) \in X^\infty$ we denote the *maximal common prefix* of two different words $w_1, w_2 \in X^\infty$, i.e., the finite word $w \in X^*$ for which there are two words $w'_1, w'_2 \in X^\infty$ such that $w_1 = w \cdot w'_1$, $w_2 = w \cdot w'_2$, and $\text{fst}(w'_1) \neq \text{fst}(w'_2)$. By convention, we set $\text{pfX}(w, w) \triangleq w$.

Trees: For a set Δ of objects, called *directions*, a Δ -*tree* is a set $T \subseteq \Delta^*$ closed under prefix, i.e., if $t \cdot d \in T$, with $d \in \Delta$, then also $t \in T$. We say that it is *complete* if it holds that $t \cdot d' \in T$ whenever $t \cdot d \in T$, for all $d' < d$, where $< \subseteq \Delta \times \Delta$ is an a priori fixed strict total order on the set of directions that is clear from the context. Moreover, it is *full* if $T = \Delta^*$. The elements of T are called *nodes* and the empty word ε is the *root* of T . For every $t \in T$ and $d \in \Delta$, the node $t \cdot d \in T$ is a *successor* of t in T . The tree is *b-bounded* if the maximal number b of its successor nodes is finite, i.e., $b = \max_{t \in T} |\{t \cdot d \in T : d \in \Delta\}| < \omega$. A *branch* of the tree is an infinite word $w \in \Delta^\omega$ such that $(w)_{\leq i} \in T$, for all $i \in \mathbb{N}$. For a finite set Σ of objects, called *symbols*, a Σ -*labeled* Δ -*tree* is a quadruple $\langle \Sigma, \Delta, T, \nu \rangle$, where T is a Δ -tree and $\nu : T \rightarrow \Sigma$ is a *labeling function*. When Δ and Σ are clear from the context, we call $\langle T, \nu \rangle$ simply a (labeled) tree.

B. Expressiveness

In this appendix, we report the result about the variance of $\text{SL}[XG]$ w.r.t. the decision-unwinding. This fact, together with the invariance of $\text{SL}[IG]$ under the same kind of unwinding [19], implies that the latter, and consequently ATL^* , are strictly less expressive than $\text{SL}[XG]$.

To do this, we first introduce two particular kinds of CGS whose structure is a directed tree.

Definition A.1 (Concurrent Game Trees). A concurrent game tree (CGT, for short) is a CGS $\mathcal{T} \triangleq \langle \text{AP}, \text{Ag}, \text{Ac}, \text{St}, \lambda, \tau, \varepsilon \rangle$, where (i) $\text{St} \subseteq \Delta^*$ is a Δ -tree for a given set Δ of directions and (ii) if $t \cdot e \in \text{St}$ then there is a decision $d \in \text{Dc}$ such that $\tau(t, d) = t \cdot e$, for all $t \in \text{St}$ and $e \in \Delta$. Furthermore, \mathcal{T} is a decision tree (DT, for short) if (i) $\text{St} = \text{Dc}^*$ and (ii) if $t \cdot d \in \text{St}$ then $\tau(t, d) = t \cdot d$, for all $t \in \text{St}$ and $d \in \text{Dc}$.

Now, we can report the definition of decision-unwinding, which is a generalization for CGSS of the classic concept of *unwinding* of labeled transition systems.

Definition A.2 (Decision-Unwinding). *Let \mathcal{G} be a CGS. Then, the decision-unwinding of \mathcal{G} is the DT $\mathcal{G}_{DU} \triangleq \langle \text{AP}, \text{Ag}, \text{Ac}_{\mathcal{G}}, \text{Dc}_{\mathcal{G}}^*, \lambda, \tau, \varepsilon \rangle$ for which there is a surjective function $\text{unw} : \text{Dc}_{\mathcal{G}}^* \rightarrow \text{St}_{\mathcal{G}}$ such that (i) $\text{unw}(\varepsilon) = s_{0\mathcal{G}}$, (ii) $\text{unw}(\tau(t, d)) = \tau_{\mathcal{G}}(\text{unw}(t), d)$, and (iii) $\lambda(t) = \lambda_{\mathcal{G}}(\text{unw}(t))$, for all $t \in \text{Dc}_{\mathcal{G}}^*$ and $d \in \text{Dc}_{\mathcal{G}}$.*

At this point, we can prove the required result.

Theorem II.1 (Unwinding Variance). *There exists an SL[XG] sentence φ and two models \mathcal{G}_1 and \mathcal{G}_2 with isomorphic decision-unwindings such that $\mathcal{G}_1 \models \varphi$ and $\mathcal{G}_2 \not\models \varphi$.*

Proof: Consider the two CGSS \mathcal{G}_1 and \mathcal{G}_2 of Figure 4, with $\text{AP} = \{p\}$, $\text{Ag} = \{\alpha, \beta\}$, $\text{Ac}_{\mathcal{G}_1} = \text{Ac}_{\mathcal{G}_2} = \{0, 1\}$, $\text{St}_{\mathcal{G}_1} = \text{St}_{\mathcal{G}_2} = \{s_0, s'_1, s''_1, s'_2, s''_2, s'_3, s''_3\}$, $\lambda_{\mathcal{G}_1} = \lambda_{\mathcal{G}_2}$, and $s_{0\mathcal{G}_1} = s_{0\mathcal{G}_2} = s_0$. It is immediate to see that they have the same decision-unwinding, i.e., $\mathcal{G}_{1DU} = \mathcal{G}_{2DU}$ (see Figure 5).

Now, consider the SL[CG] sentence $\varphi = \langle\langle x \rangle\rangle \langle\langle y_p \rangle\rangle \langle\langle y_{\neg p} \rangle\rangle ((\alpha, x)(\beta, y_p)(X X p)) \wedge ((\alpha, x)(\beta, y_{\neg p})(X X \neg p))$. It is easy to see that $\mathcal{G}_1 \not\models \varphi$, while $\mathcal{G}_2 \models \varphi$. Thus, SL[CG] cannot be invariant under decision-unwinding. Indeed, each strategy f_x of the agent α in \mathcal{G}_1 forces to reach only one state at a time among s'_2, s''_2, s'_3 , and s''_3 . Formally, for each strategy $f_x \in \text{Str}_{\mathcal{G}_1}(s_0)$, there is a state $s \in \{s'_2, s''_2, s'_3, s''_3\}$ such that, for all strategies $f_y \in \text{Str}_{\mathcal{G}_1}(s_0)$, it holds that $(\pi)_2 = s$, where $\pi \triangleq \text{play}(\emptyset[\alpha \mapsto f_x][\beta \mapsto f_y], s_0)$. Thus, it is impossible to satisfy both the goals $X X p$ and $X X \neg p$ with the same strategy of α .

Conversely, since s_0 in \mathcal{G}_2 is owned by the agent β , we may reach both s'_1 and s''_1 with the same strategy f_x of α . Thus, if $f_x(s_0 \cdot s'_1) \neq f_x(s_0 \cdot s''_1)$, we reach, at the same time, either the pair of states s'_2 and s''_3 or s'_2 and s'_3 . Formally, there are a strategy $f_x \in \text{Str}_{\mathcal{G}_2}(s_0)$, with $f_x(s_0 \cdot s'_1) \neq f_x(s_0 \cdot s''_1)$, a pair of states $(s_p, s_{\neg p}) \in \{(s'_2, s''_3), (s'_2, s'_3)\}$, and two strategies $f_{y_p}, f_{y_{\neg p}} \in \text{Str}_{\mathcal{G}_2}(s_0)$ such that $(\pi_p)_2 = s_p$ and $(\pi_{\neg p})_2 = s_{\neg p}$, where $\pi_p \triangleq \text{play}(\emptyset[\alpha \mapsto f_x][\beta \mapsto f_{y_p}], s_0)$ and $\pi_{\neg p} \triangleq \text{play}(\emptyset[\alpha \mapsto f_x][\beta \mapsto f_{y_{\neg p}}], s_0)$. So, we can satisfy both goals $X X p$ and $X X \neg p$ with the same strategy of α .

Finally, since the SL[DG] sentence $\varphi' = \langle\langle x \rangle\rangle \langle\langle y_p \rangle\rangle \langle\langle y_{\neg p} \rangle\rangle ((\alpha, x)(\beta, y_p)(X X p)) \vee ((\alpha, x)(\beta, y_{\neg p})(X X \neg p))$ is equivalent to $\neg \varphi$, we immediately obtain that $\mathcal{G}_1 \models \varphi'$, while $\mathcal{G}_2 \not\models \varphi'$. Therefore, SL[DG] cannot be invariant under decision-unwinding too. ■

C. Behavioral Semantics

In this appendix, we show that SL[XG] has the behavioral semantics, that is, for each $\varphi \in \text{SL[XG]}$ and CGS \mathcal{G} , it holds that $\mathcal{G} \models \varphi$ if and only if $\mathcal{G} \models_B \varphi$.

In the following, \mathcal{G} will denote a CGS, $s \in \text{St}$ a states, and $\wp \in \text{Qnt}(V)$ and $B \subseteq \text{Bnd}(V)$, respectively, a quantification

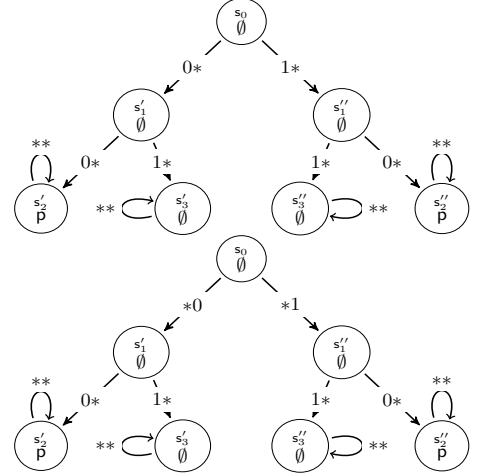


Figure 4. The two CGSS \mathcal{G}_1 and \mathcal{G}_2 with identical decision-unwinding.

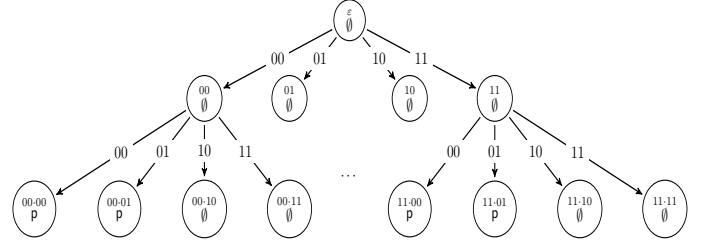


Figure 5. Part of the decision-unwinding of \mathcal{G}_1 and \mathcal{G}_2 .

prefix and a set of binding prefixes for a given set of variables $V \subseteq \text{Var}$.

1) **Two-player games:** A two-player arena (TPA, for short) is a tuple $\mathcal{A} \triangleq \langle N_e, N_o, E, n_0 \rangle$, where N_e and N_o are non-empty non-intersecting sets of nodes for players *Even* and *Odd*, respectively, $E \triangleq E_e \cup E_o$, with $E_e \subseteq N_e \times N_o$ and $E_o \subseteq N_o \times N_e$, is the edge relation between nodes, and $n_0 \in N_e$ is a designated initial node.

An *Even* (resp., *Odd*) position in \mathcal{A} is a sequence of nodes $\sigma \in (N_e \cdot N_o)^* \cdot N_e$ (resp., $\sigma \in (N_e \cdot N_o)^+ \cdot N_o$) such that $(\sigma)_0 = n_0$ and, for all $i \in [0, |\sigma| - 1]$, either $((\sigma)_i, (\sigma)_{i+1}) \in E_e$, if $i \equiv 0 \pmod{2}$, or $((\sigma)_i, (\sigma)_{i+1}) \in E_o$, otherwise. By Pos_e (resp., Pos_o) we denote the set of *Even* (resp., *Odd*) positions.

The following notions are basically the analogous of the notions of strategy, path and play we have already introduced for CGS. However, in order to avoid confusion, we use here the different names of scheme, match and match map, respectively.

An *Even* (resp., *Odd*) scheme in \mathcal{A} is a function $s_e : \text{Pos}_e \rightarrow N_o$ (resp., $s_o : \text{Pos}_o \rightarrow N_e$) that maps each *Even* (resp., *Odd*) position to an *Odd* (resp., *Even*) node in a way that is compatible with the edge relation E_e (resp., E_o), i.e., for all $\sigma \in \text{Pos}_e$ (resp., $\sigma \in \text{Pos}_o$), it holds that $(\text{lst}(\sigma), s_e(\sigma)) \in E_e$ (resp., $(\text{lst}(\sigma), s_o(\sigma)) \in E_o$). By Sch_e (resp., Sch_o) we indicate the sets of *Even* (resp., *Odd*) schemes.

A *match* in \mathcal{A} is an infinite sequence of nodes $\varpi \in (N_e \cdot N_o)^\omega$ such that $(\varpi)_0 = n_0$ and, for all $i \in \mathbb{N}$,

either $((\varpi)_i, (\varpi)_{i+1}) \in E_e$, if $i \equiv 0 \pmod{2}$, or $((\varpi)_i, (\varpi)_{i+1}) \in E_o$, otherwise. By Mtc we denote the set of all matches. The *match map* $\text{mtc} : \text{Sch}_e \times \text{Sch}_o \rightarrow \text{Mtc}$ is the function that, given two schemes $s_e \in \text{Sch}_e$ and $s_o \in \text{Sch}_o$, returns the unique match $\varpi = \text{mtc}(s_e, s_o)$ such that, for all $i \in \mathbb{N}$, it holds that either $(\varpi)_{i+1} = s_e((\varpi)_{\leq i})$, if $i \equiv 0 \pmod{2}$, or $(\varpi)_{i+1} = s_o((\varpi)_{\leq i})$, otherwise.

A *two-player game* (TPG, for short) is a tuple $\mathcal{H} \triangleq \langle \mathcal{A}, \text{Win} \rangle$, where \mathcal{A} is a TPA and $\text{Win} \subseteq \text{Mtc}$ is the *winning set*, i.e., the set of all matches that are winning for Even. We say that player Even (resp., Odd) wins \mathcal{H} , i.e., he has a winning scheme, if there exists an Even (resp., Odd) scheme $s_e \in \text{Sch}_e$ (resp., $s_o \in \text{Sch}_o$) such that, for all Odd (resp., Even) schemes $s_o \in \text{Sch}_o$ (resp., $s_e \in \text{Sch}_e$), it holds that $\text{mtc}(s_e, s_o) \in \text{Win}$ (resp., $\text{mtc}(s_e, s_o) \notin \text{Win}$).

2) **Topological definitions:** We provide some preliminary definitions about set of walks. These definitions formalize the type of set of walks that derive from conjunctive goals.

Definition A.3 (Open Walk Sets). *Let $W \subseteq \text{Wlk}_{\mathcal{G}}(\mathcal{B}, s)$ be a set of walks. Then, W is open if, for each $\text{wlk} \in W$, there exists $b \in \mathcal{B}$ such that, for all $\text{wlk}' \in \text{Wlk}_{\mathcal{G}}(\mathcal{B}, s)$, if $\text{wlk}'(b) = \text{wlk}(b)$ then $\text{wlk}' \in W$.*

Intuitively, a set of walks W is open if it represents for each walk wlk a disjunctive condition over its paths $\text{wlk}(b)$.

Definition A.4 (Closed Walk Sets). *Let $W \subseteq \text{Wlk}_{\mathcal{G}}(\mathcal{B}, s)$ be a set of walks. Then, W is closed if, for each $\text{wlk} \in \text{Wlk}_{\mathcal{G}}(\mathcal{B}, s)$, it holds that, if for all $b \in \mathcal{B}$ there exists a $\text{wlk}' \in W$ such that $\text{wlk}(b) = \text{wlk}'(b)$ then $\text{wlk} \in W$.*

Intuitively, a set of walks W is closed if we collect the single paths $\text{wlk}(b)$ with $\text{wlk} \in W$ and then recombine them. This means that W represents for each walk wlk a conjunctive condition over its paths $\text{wlk}(b)$.

Lemma A.1 (Open vs Closed Walk Set). *Let $W \subseteq \text{Wlk}_{\mathcal{G}}(\mathcal{B}, s)$ be a set of walks. Then, W is open iff $\text{Wlk}_{\mathcal{G}}(\mathcal{B}, s) \setminus W$ is closed.*

The following lemma formally shows the intuition behind open and closed sets of walks.

Lemma A.2 (SL[XG] Topology). *Let ψ_b be an SL formula with $\text{free}(\psi_b) = \text{Ag}$, for each binding $b \in \mathcal{B}$. Then, the following holds:*

- 1) *the set $\{\text{wlk}_{\mathcal{G}}(\chi, \mathcal{B}, s) : \chi \in \text{Asg}_{\mathcal{G}}(\mathcal{V}, s) \wedge \mathcal{G}, \chi, s \models \bigvee_{b \in \mathcal{B}} b\psi_b\}$ is open;*
- 2) *the set $\{\text{wlk}_{\mathcal{G}}(\chi, \mathcal{B}, s) : \chi \in \text{Asg}_{\mathcal{G}}(\mathcal{V}, s) \wedge \mathcal{G}, \chi, s \models \bigwedge_{b \in \mathcal{B}} b\psi_b\}$ is closed.*

3) **Decoupled schemes:** The proof of elementariness for SL[XG] uses some general properties of functions that are formalized in the following definitions.

Definition A.5 (Decoupled Functions). *Let X, Y , and Z be three sets and $f : (X \rightarrow Y) \rightarrow (X \rightarrow Z)$ a function.*

Then, f is decoupled if, for all elements $x \in X$ and functions $g_1, g_2 : X \rightarrow Y$ with $g_1(x) = g_2(x)$, it holds that $f(g_1)(x) = f(g_2)(x)$.

Intuitively, the previous definition says that the behavior of $f(g)$ on x does not depend on the whole g , but only on the value $g(x)$.

Definition A.6 (Connection Functions). *Let X, Y , and Z be three sets and $f : (X \rightarrow Y) \rightarrow (X \rightarrow Z)$ and $\tilde{f} : X \rightarrow (Y \rightarrow Z)$ two functions. Then, \tilde{f} is the connection of f if, for all $x \in X$ and $g : X \rightarrow Y$, it holds that $\tilde{f}(x)(g(x)) = f(g)(x)$.*

As shown by the following lemma, Definitions A.5 and A.6 have actually the same meaning.

Lemma A.3 (Decoupled Function). *Let X, Y , and Z be three sets and $f : (X \rightarrow Y) \rightarrow (X \rightarrow Z)$ a function. Then, f is decoupled iff there exists a connection function for it.*

Definition A.7 (Simple Function Sets). *Let $S \subseteq X \rightarrow Y$ be a set of functions between two sets X and Y . Moreover, let $S_{\downarrow x} \triangleq \{g(x) : g \in S\}$, for all $x \in X$. Then, S is simple if, for all $g : X \rightarrow Y$, it holds that if $g(x) \in S_{\downarrow x}$, for all $x \in X$, then $g \in S$.*

Simple sets represent a generalization over generic functions of the notion of closed sets of walks.

Lemma A.4 (Simple Function Set). *Let X, Y , and Z be three sets and $f : (X \rightarrow Y) \rightarrow (X \rightarrow Z)$ a decoupled function. Then, $\text{rng}(f)$ is simple.*

Proof: Let $S \triangleq \text{rng}(f)$. Then, to prove the statement, we have to show that, given a generic function $h : X \rightarrow Z$, if $h(x) \in S_{\downarrow x}$, for all $x \in X$, then $h \in S$. So, for each $x \in X$, suppose that $h(x) \in S_{\downarrow x}$. By definition of the set $S_{\downarrow x}$, there is a function $h_x \in S$ such that $h(x) = h_x(x)$. Therefore, since S is the range of f , there exists a function $g_x : X \rightarrow Y$ such that $h(x) = f(g_x)(x)$. Now, let $g : X \rightarrow Y$ be such that $g(x) = g_x(x)$, for all $x \in X$. Then, due to the fact that f is decoupled, we obtain that $h(x) = f(g_x)(x) = f(g)(x)$, for all $x \in X$. Hence, $h = f(g)$, which implies that $h \in S$. ■

We now give the definition of what we mean for decoupled Odd and Even schemes.

Definition A.8 (Decoupled Odd Schemes). *Let \mathcal{G} be a CGS, $s \in \text{St}$ one of its states, and $\wp \in \text{Qnt}(\mathcal{V})$ and $\mathcal{B} \subseteq \text{Bnd}(\mathcal{V})$, respectively, a quantification prefix and a set of binding prefixes for a given set of variables $\mathcal{V} \subseteq \text{Var}$. Then, an Odd scheme $s_o \in \text{Sch}_o$ in the TPA $\mathcal{A}(\mathcal{G}, s, \wp, \mathcal{B})$ is decoupled if, for all Even positions $\sigma \in \text{Pos}_e$, the function $f_{\sigma} : \text{DC}_{\mathcal{G}}(\wp, \mathcal{B}, \text{lst}(\sigma), s) \rightarrow \text{Evl}_{\mathcal{G}}(\mathcal{B}, s)$ defined as follows is decoupled: $f_{\sigma}(\text{dc}) \triangleq s_o(\sigma \cdot (\text{lst}(\sigma), \text{dc}))$, for all $\text{dc} \in \text{DC}_{\mathcal{G}}(\wp, \mathcal{B}, \text{lst}(\sigma), s)$.*

Intuitively, Definition A.8 simply says that an Odd scheme is a decoupled function with respect to the dependence maps chosen by the Even player.

Definition A.9 (Decoupled Even Schemes). *Let \mathcal{G} be a CGS, $s \in \text{St}$ one of its states, and $\wp \in \text{Qnt}(V)$ and $B \subseteq \text{Bnd}(V)$, respectively, a quantification prefix and a set of binding prefixes for a given set of variables $V \subseteq \text{Var}$. Then, an Even scheme $s_e \in \text{Sch}_e$ in the TPA $\mathcal{A}(\mathcal{G}, s, \wp, B)$ is decoupled if, for all Odd positions $\sigma \in \text{Pos}_o$, the function $f_\sigma : \{\text{evl} \in \text{Evl}_{\mathcal{G}}(B, s) : \sigma \cdot \text{evl} \in \text{Pos}_e\} \rightarrow \text{DM}_{\text{Ac}_{\mathcal{G}}}(\wp)^B$ satisfying the following property is decoupled: $s_e(\sigma \cdot \text{evl}) = (\text{evl}, f_\sigma(\text{evl}))$, for all $\text{evl} \in \text{dom}(f_\sigma)$.*

Mutandum mutandis, Definition A.9 has the same meaning as Definition A.8.

Lemma A.5 (Decoupled Odd Scheme). *Let \mathcal{G} be a CGS, $s \in \text{St}$ one of its states, $\wp \in \text{Qnt}(V)$ and $B \subseteq \text{Bnd}(V)$, respectively, a quantification prefix and a set of binding prefixes for a given set of variables $V \subseteq \text{Var}$, and $W \subseteq \text{Wlk}_{\mathcal{G}}(B, s)$ an open/closed set of walks. Moreover, let $\mathcal{H} = \langle \mathcal{A}, \text{Win} \rangle = \mathcal{H}(\mathcal{G}, s, W, \wp, B)$ be the dependence-vs-valuation game for \mathcal{G} in s over W w.r.t. \wp and B . Then, if player Odd has a winning scheme in \mathcal{H} , he has a decoupled winning scheme.*

Proof: The following proof regards the case of open sets of walks only. The case of closed sets can be proved in a similar way.

To prove the statement, we show how to transform a given winning Odd scheme s_o in a decoupled one s_o' . To do this, we first define, in a mutual recursive way, two functions through which we map, respectively, positions and decision clusters on which to define the function s_o' into those on which we can evaluate the function s_o , for each binding in the set B . Then, we prove that the obtained scheme s_o' is winning, by using the assumption on the set of walks W .

Let $P \triangleq \{(\sigma, C) \in \text{Pos}_e \times 2^B : C \in (B/\equiv_{\text{lst}(\sigma)})\}$ be the set of pairs of Even positions and related classes of bindings. Then, it is easy to see that, $(n_0, B) \in P$ and, for each pair $(\sigma \cdot (\text{lst}(\sigma), \text{dc}) \cdot \text{evl}, C) \in P$, there is a unique class of bindings $C' \in (B/\equiv_{\text{lst}(\sigma)})$ such that $C \subseteq C'$. By $c : P \setminus \{(n_0, B)\} \rightarrow 2^B$ we indicate the function returning such a class, i.e., $C \subseteq c(\sigma \cdot (\text{lst}(\sigma), \text{dc}) \cdot \text{evl}, C)$ and $(\sigma, c(\sigma \cdot (\text{lst}(\sigma), \text{dc}) \cdot \text{evl}, C)) \in P$, for all $(\sigma \cdot (\text{lst}(\sigma), \text{dc}) \cdot \text{evl}, C) \in P$. Observe that $c(\sigma \cdot (\text{lst}(\sigma), \text{dc}) \cdot \text{evl}, [b]_{\text{evl}}) = [b]_{\text{lst}(\sigma)}$, for all $b \in B$. Furthermore, let $Q \triangleq \{(\sigma, C, \theta) \in \text{Pos}_e \times 2^B \times \text{DM}_{\text{Ac}_{\mathcal{G}}}(\wp) : (\sigma, C) \in P\}$ be the set of triples containing Even positions, related classes of bindings, and dependence maps.

Now, consider two functions $p : P \rightarrow \text{Pos}_e$ and $q : Q \rightarrow \text{DM}_{\text{Ac}_{\mathcal{G}}}(\wp)^B$ satisfying the following mutual recursive properties.

- 1) a) $p(n_0, B) = n_0$.
- b) For all $(\sigma \cdot (\text{lst}(\sigma), \text{dc}) \cdot \text{evl}, C) \in P$, if $\text{evl}|_C = \text{evl}'|_C$ then $p(\sigma \cdot (\text{lst}(\sigma), \text{dc}) \cdot \text{evl}, C) = \sigma' \cdot (\text{lst}(\sigma'), \text{dc}') \cdot \text{evl}'$, where the Even position σ' , the dependence cluster dc' , and the evolution evl' are defined as follows:

- $\sigma' \triangleq p(\sigma, c(\sigma \cdot (\text{lst}(\sigma), \text{dc}) \cdot \text{evl}, C))$;
- $\text{dc}' \triangleq q(\sigma, c(\sigma \cdot (\text{lst}(\sigma), \text{dc}) \cdot \text{evl}, C), \theta)$, where $\theta \in \text{DM}_{\text{Ac}_{\mathcal{G}}}(\wp)$ is such that $\theta = \text{dc}(b)$, for all $b \in C$;
- $\text{evl}' \triangleq s_o(\sigma' \cdot (\text{lst}(\sigma'), \text{dc}'))$.

- 2) For all $(\sigma, C, \theta) \in Q$, it holds that:
 - a) $q(\sigma, C, \theta) \in \text{DC}_{\mathcal{G}}(\wp, B, \text{lst}(p(\sigma, C)), s)$;
 - b) $q(\sigma, C, \theta)(b) = \theta$, for all $b \in C$.

At this point, we can define the new Odd scheme s_o' as follows: $s_o'(\sigma \cdot (\text{lst}(\sigma), \text{dc}))(b) \triangleq s_o(\sigma_b \cdot (\text{lst}(\sigma_b), \text{dc}_b))(b)$, where $\sigma_b \triangleq p(\sigma, [b]_{\text{lst}(\sigma)})$ and $\text{dc}_b \triangleq q(\sigma, [b]_{\text{lst}(\sigma)}, \text{dc}(b))$, for all $\sigma \cdot (\text{lst}(\sigma), \text{dc}) \in \text{Pos}_o$ and $b \in B$. It is easy to see that s_o' is decoupled, i.e., for each position $\sigma \in \text{Pos}_e$, the function $f_\sigma : \text{DC}_{\mathcal{G}}(\wp, B, \text{lst}(\sigma), s) \rightarrow \text{Evl}_{\mathcal{G}}(B, \text{lst}(\sigma))$ is decoupled as well, where $f_\sigma(\text{dc}) \triangleq s_o'(\sigma \cdot (\text{lst}(\sigma), \text{dc}))$, for all $\text{dc} \in \text{DC}_{\mathcal{G}}(\wp, B, \text{lst}(\sigma), s)$. To show this fact, let $b \in B$ be a binding and $\text{dc}_1, \text{dc}_2 \in \text{DC}_{\mathcal{G}}(\wp, B, \text{lst}(\sigma), s)$ two dependence clusters such that $\text{dc}_1(b) = \text{dc}_2(b)$. Then, it is immediate to see that $f_\sigma(\text{dc}_1)(b) = s_o'(\sigma \cdot (\text{lst}(\sigma), \text{dc}_1))(b) = s_o(\sigma_b \cdot (\text{lst}(\sigma_b), \text{dc}_b))(b) = s_o'(\sigma \cdot (\text{lst}(\sigma), \text{dc}_2))(b) = f_\sigma(\text{dc}_2)(b)$, where $\text{dc}_b = q(\sigma, [b]_{\text{lst}(\sigma)}, \text{dc}_1(b)) = q(\sigma, [b]_{\text{lst}(\sigma)}, \text{dc}_2(b))$.

It remains to prove that s_o' is a winning scheme for the Odd player, i.e. for all Even scheme $s_e \in \text{Sch}_e$, it holds that $\varpi \triangleq \text{mtc}(s_e, s_o') \notin \text{Win}$, which means that $\varpi \in \text{Mtc} \setminus \text{Win}$. To do this, fix an Even scheme $s_e \in \text{Sch}_e$. We now proceed by proving in sequence the following five steps, for each binding $b \in B$.

- 1) $((\varpi)_{2i})_{\uparrow [b]_{(\varpi)_{2i}}} = ((p((\varpi)_{\leq 2i}, [b]_{(\varpi)_{2i}}))_{2i})_{\uparrow [b]_{(\varpi)_{2i}}}$, for all $i \in \mathbb{N}$.
- 2) $p((\varpi)_{\leq 2j}, [b]_{(\varpi)_{2j}}) = (p((\varpi)_{\leq 2i}, [b]_{(\varpi)_{2i}}))_{\leq 2j}$, for all $i, j \in \mathbb{N}$, with $j \leq i$.
- 3) There is a unique match $\varpi_b \in \text{Mtc}$ such that $(\varpi_b)_{\leq 2i} = p((\varpi)_{\leq 2i}, [b]_{(\varpi)_{2i}})$, for all $i \in \mathbb{N}$.
- 4) There is an Even scheme $s_{e_b} \in \text{Sch}_e$ such that (i) $\varpi_b = \text{mtc}(s_{e_b}, s_o)$ and (ii) $s_{e_b}((\varpi_b)_{\leq 2i}) = ((\varpi_b)_{2i}, q((\varpi)_{\leq 2i}, [b]_{(\varpi)_{2i}}, \text{dc}(b)))$, where the dependence cluster $\text{dc} \in \text{DC}_{\mathcal{G}}(\wp, B, (\varpi)_{2i}, s)$ is such that $s_e((\varpi)_{\leq 2i}) = ((\varpi)_{2i}, \text{dc})$, for all $i \in \mathbb{N}$.
- 5) $(\varpi)_{2i}(b) = (\varpi_b)_{2i}(b)$, for all $i \in \mathbb{N}$.

To prove the first item, observe that $(\varpi)_0 = n_0 = p(n_0, B)$. Thus, for $i = 0$, the thesis is immediate. Now, fix $i > 0$ and let $\text{dc} \in \text{DC}_{\mathcal{G}}(\wp, B, (\varpi)_{2(i-1)}, s)$ be the dependence cluster such that $(\varpi)_{\leq 2i} = (\varpi)_{\leq 2(i-1)} \cdot ((\varpi)_{2(i-1)}, \text{dc}) \cdot (\varpi)_{2i}$. Then, it holds that $((\varpi)_{2i})_{\uparrow [b]_{(\varpi)_{2i}}} = s_o'((\varpi)_{\leq 2(i-1)} \cdot ((\varpi)_{2(i-1)}, \text{dc}))_{\uparrow [b]_{(\varpi)_{2i}}} = s_o(\sigma_b \cdot (\text{lst}(\sigma_b), \text{dc}_b))_{\uparrow [b]_{(\varpi)_{2i}}}$, where $\sigma_b \triangleq p((\varpi)_{\leq 2(i-1)}, [b]_{(\varpi)_{2(i-1)}})$ and $\text{dc}_b \triangleq q((\varpi)_{\leq 2(i-1)}, [b]_{(\varpi)_{2(i-1)}}, \text{dc}(b))$. Now, by definition of the function p , we derive that $(p((\varpi)_{\leq 2i}, [b]_{(\varpi)_{2i}}))_{2i} = s_o(\sigma_b \cdot (\text{lst}(\sigma_b), \text{dc}_b))$. Hence, the thesis follows immediately.

The second item can be easily proved by induction on $i \in \mathbb{N}$, for all $j \in \mathbb{N}$ with $j \leq i$. Let $\sigma_i \triangleq p((\varpi)_{\leq 2i}, [b]_{(\varpi)_{2i}})$. As base case, $i = 0$. Consequently, $j = 0$. So, we immediately

obtain that $\sigma_j = \sigma_0 = (\sigma_0)_{\leq 0} = (\sigma_i)_{\leq j}$. As inductive case, for a given $i \in \mathbb{N}$, suppose that $\sigma_j = (\sigma_i)_{\leq j}$, for all $j \in \mathbb{N}$ with $j \leq i$. We now prove that the same holds for $i + 1$. If $j = i + 1$, the statement is trivial, since $\sigma_j = \sigma_{i+1} = (\sigma_{i+1})_{\leq i+1} = (\sigma_{i+1})_{\leq j}$. Therefore, suppose that $j \leq i$. By definition of the function p , the previous item, and the fact that $c((\varpi)_{\leq 2(i+1)}, [b]_{(\varpi)_{2(i+1)}}) = [b]_{(\varpi)_{2i}}$, we have that $\sigma_{i+1} = \sigma_i \cdot (\text{lst}(\sigma_i), \text{dc}') \cdot \text{evl}'$, where the decision cluster dc' and the evolution evl' are derived from $(\varpi)_{2(i+1)}$ via the functions p and q . Therefore, by the inductive hypothesis, we obtain that $(\sigma_{i+1})_{\leq 2j} = (\sigma_i)_{\leq 2j} = \sigma_j$. So, the thesis is proved.

For the third item, let $\varpi_b \in (\mathbb{N}_e \times \mathbb{N}_o)^\omega$ be the infinite word such that (i) $(\varpi_b)_0 \triangleq n_0$, (ii) $(\varpi_b)_{2i-1} \triangleq (p((\varpi)_{\leq 2i}, [b]_{(\varpi)_{2i}}))_{2i-1}$, and (iii) $(\varpi_b)_{2i} \triangleq (p((\varpi)_{\leq 2i}, [b]_{(\varpi)_{2i}}))_{2i}$, for all $i \in [1, \omega[$. It is easy to see that ϖ_b is a match. Now, again by induction on $i \in \mathbb{N}$, we prove that $(\varpi_b)_{\leq 2i} = p((\varpi)_{\leq 2i}, [b]_{(\varpi)_{2i}})$. The case base $i = 0$ is immediate by definition. So, let $i > 0$ and suppose, by inductive hypothesis, that $(\varpi_b)_{\leq 2(i-1)} = p((\varpi)_{\leq 2(i-1)}, [b]_{(\varpi)_{2(i-1)}})$. Then, by the previous item, we have that $(\varpi_b)_{\leq 2(i-1)} = (p((\varpi)_{\leq 2i}, [b]_{(\varpi)_{2i}}))_{2(i-1)}$. Consequently, by the choice of ϖ_b , it holds that $(\varpi_b)_{\leq 2i} = (p((\varpi)_{\leq 2i}, [b]_{(\varpi)_{2i}}))_{2(i-1)} \cdot (p((\varpi)_{\leq 2i}, [b]_{(\varpi)_{2i}}))_{2i-1} \cdot (p((\varpi)_{\leq 2i}, [b]_{(\varpi)_{2i}}))_{2i} = p((\varpi)_{\leq 2i}, [b]_{(\varpi)_{2i}})$.

To prove the fourth item, let $s_{e_b} \in \text{Sch}_e$ be an Even scheme such that $s_{e_b}((\varpi_b)_{\leq 2i}) = ((\varpi_b)_{2i}, q((\varpi)_{\leq 2i}, [b]_{(\varpi)_{2i}}, \text{dc}(b)))$, for all $i \in \mathbb{N}$, where the dependence cluster $\text{dc} \in \text{DC}_{\mathcal{G}}(\wp, B, (\varpi)_{2i}, s)$ is derived by the second component of the scheme $s_e((\varpi)_{\leq 2i}) = ((\varpi)_{2i}, \text{dc})$. The existence of such a scheme is ensured by the definition of the function q and the previous item, since $q((\varpi)_{\leq 2i}, [b]_{(\varpi)_{2i}}, \text{dc}(b)) \in \text{DC}_{\mathcal{G}}(\wp, B, \text{lst}(p((\varpi)_{\leq 2i}, [b]_{(\varpi)_{2i}})), s) = \text{DC}_{\mathcal{G}}(\wp, B, (\varpi_b)_{2i}, s)$. At this point, it remains to show that $\varpi_b = \text{mtc}(s_{e_b}, s_o)$. To do this, let $\varpi'_b = \text{mtc}(s_{e_b}, s_o)$. By a last induction on $i \in \mathbb{N}$, we prove that $(\varpi_b)_{\leq 2i} = (\varpi'_b)_{\leq 2i}$. The case base $i = 0$ is immediate by definition. So, let $i > 0$ and suppose, by inductive hypothesis, that $(\varpi_b)_{\leq 2(i-1)} = (\varpi'_b)_{\leq 2(i-1)}$. First observe that $(\varpi)_{\leq 2i} = (\varpi)_{\leq 2(i-1)} \cdot s_e((\varpi)_{\leq 2(i-1)}) \cdot s_o'((\varpi)_{\leq 2(i-1)}) \cdot s_e((\varpi)_{\leq 2(i-1)})$, since $\varpi \triangleq \text{mtc}(s_e, s_o')$. Therefore, by the first and third item, the definition of the function p , and the fact that $c((\varpi)_{\leq 2i}, [b]_{(\varpi)_{2i}}) = [b]_{(\varpi)_{2(i-1)}}$, we have that $(\varpi_b)_{\leq 2i} = p((\varpi)_{\leq 2i}, [b]_{(\varpi)_{2i}}) = (\varpi_b)_{\leq 2(i-1)} \cdot ((\varpi_b)_{2(i-1)}, \text{dc}') \cdot s_o((\varpi_b)_{\leq 2(i-1)}, ((\varpi_b)_{2(i-1)}, \text{dc}'))$, where $\text{dc}' = q((\varpi)_{\leq 2(i-1)}, [b]_{(\varpi)_{2(i-1)}}, \text{dc}(b))$ is obtained from the dependence cluster dc contained in the scheme $s_e((\varpi)_{\leq 2(i-1)}) = ((\varpi)_{2(i-1)}, \text{dc})$. Now, by the choice of the scheme s_{e_b} , it holds that $(\varpi_b)_{\leq 2i} = (\varpi_b)_{\leq 2(i-1)} \cdot s_{e_b}((\varpi_b)_{\leq 2(i-1)}) \cdot s_o((\varpi_b)_{\leq 2(i-1)}) \cdot s_{e_b}((\varpi_b)_{\leq 2(i-1)})$. Hence, by the inductive hypothesis and the definition of ϖ'_b , we

have that $(\varpi_b)_{\leq 2i} = (\varpi'_b)_{\leq 2(i-1)} \cdot s_{e_b}((\varpi'_b)_{\leq 2(i-1)}) \cdot s_o((\varpi'_b)_{\leq 2(i-1)}) \cdot s_{e_b}((\varpi'_b)_{\leq 2(i-1)}) = (\varpi'_b)_{\leq 2i}$.

We finally show the fifth item. For $i = 0$, it is immediate to see that, for all $b \in B$, it holds that $(\varpi)_0(b) = (\varpi_b)_0(b)$, since $(\varpi)_0 = n_0 = (\varpi_b)_0$. So, suppose that $i > 0$. By definition of ϖ , it holds that $(\varpi)_{2i} = s_o'((\varpi)_{\leq 2(i-1)}) \cdot s_e((\varpi)_{\leq 2(i-1)})$. Therefore, by definition of s_o' , we have that $(\varpi)_{2i}(b) = s_o'((\varpi)_{\leq 2(i-1)}) \cdot s_e((\varpi)_{\leq 2(i-1)})(b) = s_o(\sigma_b \cdot (\text{lst}(\sigma_b), \text{dc}_b))(b)$, where the Even position σ_b and the dependence cluster dc_b are defined as follows: $\sigma_b \triangleq p((\varpi)_{\leq 2(i-1)}, [b]_{(\varpi)_{2(i-1)}})$ and $\text{dc}_b \triangleq q((\varpi)_{\leq 2(i-1)}, [b]_{(\varpi)_{2(i-1)}}, \text{dc}(b))$ with $s_e((\varpi)_{\leq 2(i-1)}) = ((\varpi)_{2(i-1)}, \text{dc})$. Now, by the third item and part (ii) of the fourth one, we obtain that $\sigma_b = (\varpi_b)_{\leq 2(i-1)}$ and $(\text{lst}(\sigma_b), \text{dc}_b) = s_{e_b}((\varpi_b)_{\leq 2(i-1)})$. Thus, $(\varpi)_{2i}(b) = s_o((\varpi_b)_{\leq 2(i-1)}) \cdot s_{e_b}((\varpi_b)_{\leq 2(i-1)})(b)$. Hence, by part (i) of the fourth item, we derive that $(\varpi)_{2i}(b) = (\varpi_b)_{2i}(b)$.

At this point, since s_o is a winning scheme, by part (i) of the fourth item, we obtain that $\varpi_b \in \text{Mtc} \setminus \text{Win} = \{\varpi' \in \text{Mtc} : \exists \text{wlk} \in W' \cdot \forall i \in \mathbb{N}, b \in B. (\varpi')_{2i}(b) = (\text{wlk}(b))_i\}$, where $W' \triangleq \text{Wlk}_{\mathcal{G}}(B, s) \setminus W$, for all $b \in B$. Now, due to the fact that W is an open/closed set, by Lemma A.1 on open vs closed walk set, we have that W' is a closed/open set. Therefore, for each $\varpi' \in \text{Mtc}$, it holds that, if for all $b \in B$ there is a $\varpi'' \in \text{Mtc} \setminus \text{Win}$ such that $(\varpi')_{2i}(b) = (\varpi'')_{2i}(b)$, for all $i \in \mathbb{N}$, then $\varpi' \in \text{Mtc} \setminus \text{Win}$. Hence, by the last item, we derive that $\varpi \in \text{Mtc} \setminus \text{Win}$. ■

We end this section by giving a Lemma specular to above one for Even scheme. Consequently, the proof is an easily adaptation of that lemma and, therefore, it is omitted.

Lemma A.6 (Decoupled Even Scheme). *Let \mathcal{G} be a CGS, $s \in \text{St}$ one of its states, $\wp \in \text{Qnt}(V)$ and $B \subseteq \text{Bnd}(V)$, respectively, a quantification prefix and a set of binding prefixes for a given set of variables $V \subseteq \text{Var}$, and $W \subseteq \text{Wlk}_{\mathcal{G}}(B, s)$ an open/closed set of walks. Moreover, let $\mathcal{H} = \langle \mathcal{A}, \text{Win} \rangle = \mathcal{H}(\mathcal{G}, s, W, \wp, B)$ be the dependence-vs-valuation game for \mathcal{G} in s over W w.r.t. \wp and B . Then, if player Even has a winning scheme in \mathcal{H} , he has a decoupled winning scheme.*

4) Encasement characterization: In the following lemma, we give a characterization for the winning conditions of the two players via the encasement property. In particular, we show a one-to-one relationship between the winning in the dependence-vs-valuation game of player Even and the verification of the encasement property of the associated walk set. Moreover, in the case that the winning condition of the latter is a Borelian set, by means of Martin's Determinacy Theorem, we can complete the characterization by also proving the existence of a one-to-one relationship between the winning of player Odd and the negation of the encasement property.

Lemma A.7 (Encasement Characterization). *Let $W \subseteq$*

$\text{Wlk}_{\mathcal{G}}(\mathbb{B}, s)$ be an open/closed set of walks and $\mathcal{H} = \langle \mathcal{A}, \text{Win} \rangle = \mathcal{H}(\mathcal{G}, s, W, \wp, \mathbb{B})$ be the dependence-vs-valuation game for \mathcal{G} in s over W w.r.t. \wp and \mathbb{B} . Then, the following hold:

- 1) if player Even wins \mathcal{H} then W is an encasement w.r.t. \wp and \mathbb{B} ;
- 2) if Win is a Borelian set and W is not an encasement w.r.t. \wp and \mathbb{B} then player Odd wins \mathcal{H} .

Proof: [Item 1]. Suppose that player Even wins the TPG \mathcal{H} . Then, by Lemma A.6, there exists a decoupled Even scheme $s_e \in \text{Sch}_e$ such that, for all Odd schemes $s_o \in \text{Sch}_o$, it holds that $\text{mtc}(s_e, s_o) \in \text{Win}$. Now, to prove the statement, we have to show that there exists an elementary dependence map $\theta \in \text{EDM}_{\text{Str}(s)}(\wp)$ such that, for all assignments $\chi \in \text{Asg}(\llbracket \wp \rrbracket, s)$, it holds that $\text{wlk}_{\mathcal{G}}(\theta(\chi), \mathbb{B}, s) \in W$.

To do this, first consider, for each track $\rho \in \text{Trk}(s)$, the set $Q_\rho \triangleq \{(\sigma, b) \in \text{Pos}_e \times \mathbb{B} : |\sigma| = 2|\rho| - 1 \wedge \forall i \in [0, |\rho|] . (\sigma)_{2i}(b) = (\rho)_i\}$ of pairs of Even positions σ and bindings b such that ρ represents the projection of σ on b .

Since s_e is decoupled, for each Odd position $\sigma \in \text{Pos}_o$, there exists a decoupled function $f_\sigma : \{\text{evl} \in \text{Evl}_{\mathcal{G}}(\mathbb{B}, s) : \sigma \cdot \text{evl} \in \text{Pos}_e\} \rightarrow \text{DM}_{\text{Ac}_{\mathcal{G}}}(\wp)^{\mathbb{B}}$ such that $s_e(\sigma \cdot \text{evl}) = (\text{evl}, f_\sigma(\text{evl}))$, for all $\text{evl} \in \text{dom}(f_\sigma)$.

By an easy induction on the length of the tracks, it is possible to see that, for each $\rho \in \text{Trk}(s)$, there exists a $\theta_\rho \in \text{DM}_{\text{Ac}}(\wp)$ such that $f_\sigma(\text{evl})(b) = \theta_\rho$, for all $(\sigma \cdot \text{evl}, b) \in Q_\rho$.

At this point, let $w : \text{Trk}(s) \rightarrow \text{DM}_{\text{Ac}}(\wp)$ be the function such that $w(\rho) = \theta_\rho$, for all $\rho \in \text{Trk}(s)$. By Lemma B.3 of [18] on adjoint dependence maps, there exists an elementary dependence map $\theta \in \text{EDM}_{\text{Str}(s)}(\wp)$ for which w is the adjoint, i.e., $w = \bar{\theta}$. Moreover, let $\chi \in \text{Asg}(\llbracket \wp \rrbracket, s)$ be a generic assignment and consider a derived Odd scheme $s_o \in \text{Sch}_o$ satisfying the following property: $s_o((\sigma \cdot \text{evl}) \cdot (\text{evl}, \text{dc}))(b) = \tau(\text{evl}(b), \text{dc}(b)(\widehat{\chi}(\rho)) \circ \zeta_b)$, for all $\rho \in \text{Trk}(s)$, $(\sigma \cdot \text{evl}, b) \in Q_\rho$, and $\text{dc} \in \text{DC}_{\mathcal{G}}(\wp, \mathbb{B}, \text{evl}, s)$.

Now, since by hypothesis we have that $\varpi \triangleq \text{mtc}(s_e, s_o) \in \text{Win}$, to prove the statement we have only to show by induction that $(\text{wlk}(b))_i = (\varpi)_{2i}(b)$, for all $i \in \mathbb{N}$ and $b \in \mathbb{B}$, where $\text{wlk} = \text{wlk}(\theta(\chi), \mathbb{B}, s)$. Such an induction is easy and left to the reader.

[Item 2]. If W is closed but not an encasement w.r.t. \wp and \mathbb{B} , by Item 1, we have that player Even does not win the TPG \mathcal{H} . Now, since Win is Borelian, by the determinacy theorem, it holds that player Odd wins the same game. ■

5) Game duality: In this section we prove the fundamental relationship between dependence-vs-valuation games and their duals. Basically, we show that if the Odd player wins the game then the Even player can win the dual game. This represents the second of the two crucial steps in our elementariness proof.

Before starting, we have to introduce the following auxiliary notation related to cluster of dependence and valuation functions.

Definition A.10 (Clusters Composition). Let \mathcal{G} be a CGS, $s \in \text{St}$ one of its states, $\wp \in \text{Qnt}(\mathbb{V})$ and $\mathbb{B} \subseteq \text{Bnd}(\mathbb{V})$, respectively, a quantification prefix and a set of binding prefixes for a given set of variables $\mathbb{V} \subseteq \text{Var}$, and $\text{evl} \in \text{Evl}_{\mathcal{G}}(\mathbb{B}, s)$ an evolution. Then, by $\kappa_{\text{evl}} : \text{DC}_{\mathcal{G}}(\wp, \mathbb{B}, \text{evl}, s) \times \text{VC}_{\mathcal{G}}(\llbracket \wp \rrbracket, \mathbb{B}, \text{evl}, s) \rightarrow \text{VC}_{\mathcal{G}}(\mathbb{V}, \mathbb{B}, \text{evl}, s)$ we denote the composition function between dependence and valuation clusters such that, for all $\text{dc} \in \text{DC}_{\mathcal{G}}(\wp, \mathbb{B}, \text{evl}, s)$, $\text{vc} \in \text{VC}_{\mathcal{G}}(\llbracket \wp \rrbracket, \mathbb{B}, \text{evl}, s)$, and $b \in \mathbb{B}$, it holds that $\kappa_{\text{evl}}(\text{dc}, \text{vc})(b) = \text{dc}(b)(\text{vc}(b))$.

Definition A.11 (Clusters Transition Function). Let \mathcal{G} be a CGS, $s \in \text{St}$ one of its states, and $\mathbb{B} \subseteq \text{Bnd}(\mathbb{V})$ a set of binding prefixes for a given set of variables $\mathbb{V} \subseteq \text{Var}$. Then, by $v : \{(\text{evl}, \text{vc}) \in \text{Evl}_{\mathcal{G}}(\mathbb{B}, s) \times \text{Val}_{\text{Ac}_{\mathcal{G}}}(\mathbb{V})^{\mathbb{B}} : \text{vc} \in \text{VC}_{\mathcal{G}}(\mathbb{V}, \mathbb{B}, \text{evl}, s)\} \rightarrow \text{Evl}_{\mathcal{G}}(\mathbb{B}, s)$ we denote the cluster transition function such that, for all $\text{evl} \in \text{Evl}_{\mathcal{G}}(\mathbb{B}, s)$, $\text{vc} \in \text{VC}_{\mathcal{G}}(\llbracket \wp \rrbracket, \mathbb{B}, \text{evl}, s)$, and $b \in \mathbb{B}$, it holds that $v(\text{evl}, \text{vc})(b) = \tau(\text{evl}(b), \text{vc}(b) \circ \zeta_b)$.

We also need to state the following lemma proved to hold in [18].

Lemma A.8 (Dependence Dualization). Let $\wp \in \text{Qnt}(\mathbb{V})$ be a quantification prefix over a set of variables $\mathbb{V} \subseteq \text{Var}$, D a generic set, and $S \subseteq \text{Val}_D(\mathbb{V})$ a set of valuations of V over D . Moreover, suppose that, for all dependence maps $\theta \in \text{DM}_D(\wp)$, there is a valuation $v \in \text{Val}_D(\llbracket \wp \rrbracket)$ such that $\theta(v) \in S$. Then, there exists a dependence map $\bar{\theta} \in \text{DM}_D(\bar{\wp})$ such that, for all valuations $\bar{v} \in \text{Val}_D(\llbracket \bar{\wp} \rrbracket)$, it holds that $\bar{\theta}(\bar{v}) \in S$.

At this point, we can proceed with the duality proof.

Lemma A.9 (Dependence-vs-Valuation Duality). Let $W \subseteq \text{Wlk}_{\mathcal{G}}(\mathbb{B}, s)$ be an open/closed set of walks, $\mathcal{H} = \langle \mathcal{A}, \text{Win} \rangle = \mathcal{H}(\mathcal{G}, s, W, \wp, \mathbb{B})$ the dependence-vs-valuation game for \mathcal{G} in s over W w.r.t. \wp and \mathbb{B} , and $\bar{\mathcal{H}} = \langle \bar{\mathcal{A}}, \text{Mtc} \setminus \text{Win} \rangle = \mathcal{H}(\mathcal{G}, s, \text{Wlk}_{\mathcal{G}}(\mathbb{B}, s) \setminus W, \bar{\wp}, \mathbb{B})$ its dual game. Then, if player Odd wins the dual TPG $\bar{\mathcal{H}}$, player Even wins the TPG \mathcal{H} .

Proof: Suppose that player Odd wins the dual TPG $\bar{\mathcal{H}}$. Since W is open/closed, by Lemma A.1 on open vs closed walk set, it holds that $\text{Wlk}_{\mathcal{G}}(\mathbb{B}, s) \setminus W$ is closed/open. Therefore, by Lemma A.5 on decoupled Odd scheme, there exists a decoupled Odd scheme $\bar{s}_o \in \text{Sch}_{o, \bar{\mathcal{A}}}$ such that, for all Even schemes $\bar{s}_e \in \text{Sch}_{e, \bar{\mathcal{A}}}$, it holds that $\text{mtc}_{\bar{\mathcal{A}}}(\bar{s}_e, \bar{s}_o) \in \text{Win}$. Now, to prove that Even wins the TPG \mathcal{H} , we have to show that there exists an Even scheme $s_e \in \text{Sch}_{e, \mathcal{A}}$ such that, for all Odd schemes $s_o \in \text{Sch}_{o, \mathcal{A}}$, it holds that $\text{mtc}_{\mathcal{A}}(s_e, s_o) \in \text{Win}$.

Since \bar{s}_o is decoupled, it follows that, for all $\bar{\sigma} \in \text{Pos}_{e, \bar{\mathcal{A}}}$, the function $f_{\bar{\sigma}} : \text{DC}_{\mathcal{G}}(\bar{\wp}, \mathbb{B}, \text{lst}(\bar{\sigma}), s) \rightarrow \text{Evl}_{\mathcal{G}}(\mathbb{B}, s)$ such that $f_{\bar{\sigma}}(\bar{\text{dc}}) \triangleq s_o(\bar{\sigma} \cdot (\text{lst}(\bar{\sigma}), \bar{\text{dc}}))$, for all $\bar{\text{dc}} \in \text{DC}_{\mathcal{G}}(\bar{\wp}, \mathbb{B}, \text{lst}(\bar{\sigma}), s)$, is decoupled. By Lemma A.3 on decoupled functions, $f_{\bar{\sigma}}$ has a connection $f_{\bar{\sigma}} : \mathbb{B} \rightarrow \text{DM}_{\text{Ac}}(\bar{\wp}) \rightarrow \text{St}$. Now, it is easy to see that there exists a decoupled

function $h_{\bar{\sigma}} : DC_{\mathcal{G}}(\bar{\varphi}, B, \text{lst}(\bar{\sigma}), s) \rightarrow VC_{\mathcal{G}}(V, B, \text{lst}(\bar{\sigma}), s)$ for which its connection function $\tilde{h}_{\bar{\sigma}} : B \rightarrow DM_{Ac}(\bar{\varphi}) \rightarrow Val_{Ac}(V)$ is such that $\tilde{f}_{\bar{\sigma}}(b)(\bar{\theta}) = \tau(\text{lst}(\bar{\sigma})(b), \tilde{h}_{\bar{\sigma}}(b)(\bar{\theta}) \circ \zeta_b)$. From now on, let $S_{\bar{\sigma}} \triangleq \text{rng}(h_{\bar{\sigma}})$ and $T_{\bar{\sigma}} \triangleq \text{rng}(\tilde{f}_{\bar{\sigma}})$.

By the choice of the function $h_{\bar{\sigma}}$, we have that, for all $\bar{dc} \in DC_{\mathcal{G}}(\bar{\varphi}, B, \text{lst}(\bar{\sigma}), s)$, there exists a $\bar{vc} \in VC_{\mathcal{G}}(\llbracket \bar{\varphi} \rrbracket, B, \text{lst}(\bar{\sigma}), s)$ such that $\kappa_{\text{lst}(\bar{\sigma})}(\bar{dc}, \bar{vc}) \in S_{\bar{\sigma}}$. Therefore, it is immediate to see that, for all $b \in B$ and $\bar{\theta} \in DM_{Ac}(\bar{\varphi})$, there exists a $\bar{v} \in Val_{Ac}(\llbracket \bar{\varphi} \rrbracket)$ such that $\bar{\theta}(\bar{v}) \in S_{\bar{\sigma} \downarrow b}$. Now, by Lemma A.8 on dependence dualization, for all $b \in B$, there is a $\theta_{\bar{\sigma}} \in DM_{Ac}(\varphi)$ such that, for all $v \in Val_{Ac}(\llbracket \varphi \rrbracket)$, it holds that $\theta_{\bar{\sigma}}(v) \in S_{\bar{\sigma} \downarrow b}$. This implies that there exists a $dc_{\bar{\sigma}} \in DC_{\mathcal{G}}(\varphi, B, \text{lst}(\bar{\sigma}), s)$ such that, for all $vc \in VC_{\mathcal{G}}(\llbracket \varphi \rrbracket, B, \text{lst}(\bar{\sigma}), s)$ and $b \in B$ it holds that $\kappa_{\text{lst}(\bar{\sigma})}(dc_{\bar{\sigma}}, vc)(b) \in S_{\bar{\sigma} \downarrow b}$. At this point, since $h_{\bar{\sigma}}$ is decoupled, by Lemma A.4 on simple function set, we have that $\kappa_{\text{lst}(\bar{\sigma})}(dc_{\bar{\sigma}}, vc) \in S_{\bar{\sigma}}$.

Now, consider two partial functions $p : \text{Pos}_{eA} \rightarrow \text{Pos}_{e\bar{A}}$ and $q : \text{Pos}_{eA} \rightarrow DM_{Ac}(\bar{\varphi})^B$, with $\text{dom}(p) = \text{dom}(q)$, satisfying the following mutual recursive properties.

- 1) a) $n_0 \in \text{dom}(p)$ and $p(n_0) = n_0$.
- b) For all $\sigma \cdot (\text{lst}(\sigma), dc) \cdot \text{evl} \in \text{Pos}_{eA}$, it holds that:
 - if $\sigma \in \text{dom}(p)$ and $\text{evl} \in T_{p(\sigma)}$ then $\sigma \cdot (\text{lst}(\sigma), dc) \cdot \text{evl} \in \text{dom}(p)$ and $p(\sigma \cdot (\text{lst}(\sigma), dc) \cdot \text{evl}) = p(\sigma) \cdot (\text{lst}(p(\sigma)), q(\sigma \cdot (\text{lst}(\sigma), dc) \cdot \text{evl})) \cdot \text{evl}$;
 - if $\sigma \cdot (\text{lst}(\sigma), dc) \cdot \text{evl} \in \text{dom}(p)$ then $\sigma \in \text{dom}(p)$ and $\text{evl} \in T_{p(\sigma)}$.
- 2) For all $\sigma \cdot (\text{lst}(\sigma), dc) \cdot \text{evl} \in \text{dom}(q)$, it holds that:
 - a) $q(\sigma \cdot (\text{lst}(\sigma), dc) \cdot \text{evl}) \in DC_{\mathcal{G}}(\varphi, B, \text{lst}(p(\sigma)), s)$;
 - b) $\text{evl} = \bar{s}_0(p(\sigma) \cdot (\text{lst}(p(\sigma)), q(\sigma \cdot (\text{lst}(\sigma), dc) \cdot \text{evl})))$.

By construction, it is immediate to see that such two functions surely exist and that $\text{lst}(p(\sigma)) = \text{lst}(\sigma)$, for all $\sigma \in \text{dom}(p)$.

Now, choose an Even scheme $s_e \in \text{Sch}_{eA}$ in \mathcal{A} such that $s_e(\sigma) \triangleq (\text{lst}(\sigma), dc_{p(\sigma)})$, for all $\sigma \in \text{dom}(p)$. Moreover, let $s_o \in \text{Sch}_{oA}$ be a generic Odd scheme in \mathcal{A} and consider the derived match $\varpi \triangleq \text{mtc}_{\mathcal{A}}(s_e, s_o)$. It remains only to prove that $\varpi \in \text{Win}$. As first thing, by induction on $i \in \mathbb{N}$, it is easy to see that $(\varpi)_{\leq 2i} \in \text{dom}(p)$. Moreover, there exists an Even scheme $\bar{s}_e \in \text{Sch}_{e\bar{A}}$ in $\bar{\mathcal{A}}$ such that $\bar{s}_e(p((\varpi)_{\leq 2i}) \cdot \bar{s}_o(p((\varpi)_{\leq 2i}))) = (\varpi)_{2(i+1)}$. Therefore, $(\varpi)_{2i} = (\bar{\varpi})_{2i}$, for all $i \in \mathbb{N}$, where $\bar{\varpi} \triangleq \text{mtc}_{\bar{\mathcal{A}}}(\bar{s}_e, \bar{s}_o)$. Now, since \bar{s}_o is winning, we have that $\bar{\varpi} \in \text{Win}$. Hence, due to the structure of the winning set Win , we obtain that $\varpi \in \text{Win}$, as well. ■

6) Elementariness proof: We are finally able to prove the elementariness property for $\text{SL}[\text{XG}]$.

Theorem C.5.1 (SL[XG] Formula Elementariness). *Let \mathcal{G} be a CGS, φ an SL[XG] formula, $s \in \text{St}$ a state, and $\chi \in \text{Asg}(s)$ an s -total assignment with $\text{free}(\varphi) \subseteq \text{dom}(\chi)$. Then, it holds that $\mathcal{G}, \chi, s \models \varphi$ iff $\mathcal{G}, \chi, s \models_{\text{B}} \varphi$.*

Proof: The proof proceeds by induction on the structure

of the formula. For the sake of succinctness, we only show the most important inductive case of principal subsentences $\phi \in \text{psnt}(\varphi)$, i.e., when ϕ is of the form $\varphi \bigwedge_{b \in B} b\psi_b$, where $\varphi \in \text{Qnt}(V)$ and $B \subseteq \text{Bnd}(V)$ are, respectively, a quantification prefix and a set of binding prefixes over a set $V \subseteq \text{Var}$ of variables, and all ψ_b are SL formulas with $\text{free}(\psi_b) = \text{Ag}$.

[If]. The proof of this direction is immediate and left to the reader.

[Only if]. Assume that $\mathcal{G}, \varnothing, s \models \varphi\psi$. Then, it is easy to see that, for all elementary dependence maps $\theta \in \text{EDM}_{\text{Str}(s)}(\bar{\varphi})$, there is an assignment $\chi \in \text{Asg}(\llbracket \bar{\varphi} \rrbracket, s)$ such that $\mathcal{G}, \theta(\chi), s \models \psi$. Indeed, suppose by contradiction that there exists an elementary dependence map $\theta \in \text{EDM}_{\text{Str}(s)}(\bar{\varphi})$ such that, for all assignments $\chi \in \text{Asg}(\llbracket \bar{\varphi} \rrbracket, s)$, it holds that $\mathcal{G}, \theta(\chi), s \not\models \psi$, i.e., $\mathcal{G}, \theta(\chi), s \models \neg\psi$. Then, by Theorem II.2 of SL strategy quantification, we have that $\mathcal{G}, \varnothing, s \models \bar{\varphi}\neg\psi$, i.e., $\mathcal{G}, \varnothing, s \models \neg\varphi\psi$, and so $\mathcal{G}, \varnothing, s \not\models \varphi\psi$, which is impossible.

Now, let $W \triangleq \{\text{wlk}_{\mathcal{G}}(\chi, B, s) : \chi \in \text{Asg}_{\mathcal{G}}(V, s) \wedge \mathcal{G}, \chi, s \models \psi\}$. By Lemma A.2, we have that W is open/closed. Moreover, it is evident that, for all elementary dependence maps $\theta \in \text{EDM}_{\text{Str}(s)}(\bar{\varphi})$, there is an assignment $\chi \in \text{Asg}(\llbracket \bar{\varphi} \rrbracket, s)$ such that $\text{wlk}_{\mathcal{G}}(\theta(\chi), B, s) \notin W$.

At this point, by Definition III.2 of encasement, it is clear that W is not an encasement w.r.t. $\bar{\varphi}$ and B . In addition, since all ψ_b describe a regular language, the derived set Win is Borelian. Consequently, by Item 2 of Lemma III.1 of encasement characterization, we have that player Odd wins the TPG $\mathcal{H}(\mathcal{G}, s, W, \bar{\varphi}, B)$. Thus, by Lemma III.2 of dependence-vs-valuation duality, player Even wins the dual TPG $\mathcal{H}(\mathcal{G}, s, \text{Wlk}_{\mathcal{G}}(B, s) \setminus W, \varphi, B)$. Hence, by Item 1 of Lemma III.1, we have that $\text{Wlk}_{\mathcal{G}}(B, s) \setminus W$ is an encasement w.r.t. φ and B .

Finally, again by Definition III.2, there exists an elementary dependence map $\theta \in \text{EDM}_{\text{Str}(s)}(\varphi)$ such that, for all assignments $\chi \in \text{Asg}(\llbracket \varphi \rrbracket, s)$, it holds that $\text{wlk}_{\mathcal{G}}(\theta(\chi), B, s) \in \text{Wlk}_{\mathcal{G}}(B, s) \setminus W$.

Now, it is immediate to observe that $\text{Wlk}_{\mathcal{G}}(B, s) \setminus W = \{\text{wlk}_{\mathcal{G}}(\chi, B, s) : \chi \in \text{Asg}_{\mathcal{G}}(V, s) \wedge \mathcal{G}, \chi, s \models \psi\}$. So, by the inductive hypothesis, we have that $\text{Wlk}_{\mathcal{G}}(B, s) \setminus W = \{\text{wlk}_{\mathcal{G}}(\chi, B, s) : \chi \in \text{Asg}_{\mathcal{G}}(V, s) \wedge \mathcal{G}, \chi, s \models_{\text{B}} \psi\}$, from which we derive that there exists an elementary dependence map $\theta \in \text{EDM}_{\text{Str}(s)}(\varphi)$ such that, for all assignments $\chi \in \text{Asg}(\llbracket \varphi \rrbracket, s)$, it holds that $\mathcal{G}, \theta(\chi), s \models_{\text{B}} \psi$. Consequently, by Definition II.6 of SL elementary semantics, we have that $\mathcal{G}, \varnothing, s \models_{\text{B}} \varphi\psi$. ■

As an immediate consequence of the previous theorem, we derive the following result.

Theorem III.1 (SL[XG] Behavioral Semantics). *For all SL[XG] sentences φ , it holds that $\mathcal{G} \models \varphi$ iff $\mathcal{G} \models_{\text{B}} \varphi$.*

D. Model Checking

For the proofs of Lemmas IV.1 and IV.2, see Lemmas 5.10 and 5.12 of [18], respectively. The proof of Theorem IV.1

follows.

Theorem IV.1 (SL[XG] Sentence Automaton). *Let \mathcal{G} be a CGS, $s \in \text{St}_{\mathcal{G}}$ one of its states, and $\phi = \wp \bigwedge_{b \in B} b\psi_b$ (resp., $\phi = \wp \bigvee_{b \in B} b\psi_b$) an SL[CG] (resp., SL[DG]) sentence. Then, there exists an UCT $\mathcal{U}_{\phi}^{\mathcal{G},s}$ with $\mathcal{O}(2^{|\phi|})$ states such that $\mathcal{G}, \emptyset, s \models \phi$ iff $L(\mathcal{U}_{\phi}^{\mathcal{G},s}) \neq \emptyset$ (resp., $L(\mathcal{U}_{\phi}^{\mathcal{G},s}) = \emptyset$).*

Proof: We first prove the statement for SL[CG]. Then, for SL[DG], the result is derived by using the following observation: starting with a SL[DG] sentence ϕ , first we dualize it in an SL[CG] one $\phi' \equiv \neg\phi$ and then we apply the same construction we use for SL[CG], obtaining the UCT $\mathcal{U}_{\phi'}^{\mathcal{G},s}$. Finally, since $\mathcal{G}, \emptyset, s \models \phi'$ iff $L(\mathcal{U}_{\phi'}^{\mathcal{G},s}) \neq \emptyset$, we obtain that $\mathcal{G}, \emptyset, s \models \phi$ iff $L(\mathcal{U}_{\phi'}^{\mathcal{G},s}) = \emptyset$. We now give the construction SL[CG].

As first thing, by means of Lemma IV.1, we build the UCTs $\mathcal{U}_{b\psi_b}^{\mathcal{G}}$, for all goals $b\psi_b$ with $b \in B$, of which we made the product obtaining the new UCT $\mathcal{U}_{\bigwedge_{b \in B} b\psi_b}^{\mathcal{G}}$. It is evident that, by construction, for all states $s \in \text{St}_{\mathcal{G}}$ and assignments $\chi \in \text{Asg}_{\mathcal{G}}(\text{free}(\bigwedge_{b \in B} b\psi_b), s)$, it holds that $\mathcal{G}, \chi, s \models \bigwedge_{b \in B} b\psi_b$ iff $\mathcal{T}_{\chi} \in L(\mathcal{U}_{\bigwedge_{b \in B} b\psi_b}^{\mathcal{G}})$, where \mathcal{T}_{χ} is the assignment-state encoding for χ . At this point, by applying Lemma IV.2 to $\mathcal{U}_{\bigwedge_{b \in B} b\psi_b}^{\mathcal{G}}$, we derive the required UCT $\mathcal{U}_{\phi}^{\mathcal{G},s}$. ■