A Behavioral Hierarchy of Strategy Logic*

Fabio Mogavero, Aniello Murano, Luigi Sauro

Università degli Studi di Napoli Federico II

Abstract. Starting from the seminal work introducing *Alternating Temporal Logic*, formalisms for *strategic reasoning* have assumed a prominent role in *multi-agent systems* verification. Among the others, *Strategy Logic* (SL) allows to represent sophisticated solution concepts, by treating agent strategies as *first-order objects*.
A drawback from the high power of SL is to admit *non-behavioral strategies*: a choice of an agent, at a given point of a play, may depend on choices other agents can make in the future or in counterfactual plays. As the latter moves are unpredictable, such strategies cannot be easily implemented, making the use of the logic problematic in practice.
In this paper, we describe a hierarchy of SL fragments as syntactic restrictions of the recently defined *Boolean-Goal Strategy Logic* (SL[BG]). Specifically, we introduce *Alternating-Goal Strategy Logic* (SL[AG]) that, by imposing a suitable alternation over the nesting of the Boolean connectives in SL[BG], induces two dual chains of sets of formulas, the *conjunctive* and *disjunctive* ones. A formula belongs to the level i of the conjunctive chain if it just contains conjunctions of atomic goals together with a unique formula belonging to the disjunctive chain of level $i-1$. The disjunctive chain is defined similarly. We formally prove that classic and behavioral semantics for SL[AG] coincide. Additionally, we study the related model-checking problem showing that it is 2ExpTime-complete.

1 Introduction

In the *multi-agent system* domain, formalisms for *strategic reasoning* have assumed a prominent role in the specification, verification, and synthesis tasks [3,12–17,27, 29,31]. A story of success in this field is Alternating Temporal Logic (ATL*, for short), introduced by Alur, Henzinger, and Kupferman [3]. Such a logic has the ability to express cooperation and competition among teams of agents in order to achieve robust temporal requirements, such as *fairness*, *liveness*, etc. This is made possible thanks to the fact that ATL* formally comes as a generalization of the well known branching-time temporal logic CTL* [8], where the existential and universal path quantifiers `E` and `A` are replaced with strategic modalities of the form $\langle\!\langle \mathrm{A} \rangle\!\rangle$ and $[\![\mathrm{A}]\!]$, for a generic set A of *agents*. The simplicity of these modalities results in a "*friendly-use*" logic with an elementary complexity for

* Partially supported by the FP7 European Union project 600958-SHERPA, the Embedded System Cup Project B25B09090100007 (POR Campania FS 2007/2013, asse IV e asse V), and the OR.C.HE.S.T.R.A. MIUR PON project.

the main related decision procedures. Indeed, both the model-checking and the satisfiability problems are 2ExpTime-complete [3, 28]. On the other hand, the use of strategic modalities in ATL* is restricted in such a way that the internal quantifications are just coupled in the strict $\exists\forall$ and $\forall\exists$ alternation. Moreover, and more important, agent strategies are only treated implicitly through these modalities, so they cannot be explicitly associated with any particular agent nor used by the same agent in different contexts. All these aspects give to ATL* a number of limitations when one tries to apply it to multi-agent system reasoning and games [1, 9, 11, 20, 30].

To overcome these difficulties and, thus, be able to describe sophisticated interactions among agent behaviors, new and more powerful logics have been recently introduced [4, 6, 7, 18, 24, 32]. Among the others, *Strategy Logic* (SL), as it has been introduced in [23], allows to formalize important solution concepts by treating agent strategies as *first-order objects*. Intuitively, SL unpacks the ATL* modalities, allowing to explicitly declare the strategy profiles. Notably, strategies in SL represent general conditional plans that at each step prescribe an action on the base of the previous history. Then, by means of a binding operator, they can be liked to specific agents. This allows to reuse strategies or share them among different agents. With more details, SL makes use of the existential $\langle\langle x \rangle\rangle$ and the universal $[\![x]\!]$ strategic quantifications, which stand for *"there exists a strategy x"* and *"for all strategies x"*, respectively. Furthermore, it uses the *binding* operator (a, x) that allows to bind an agent a to the strategy associated with a variable x. Using these operators, key game-theoretic properties such as *Nash equilibria* and *sub-game perfect equilibria*, not expressible in ATL*, can be described in SL.

Apart from the expressive gain of SL with respect to ATL*, the finer-grained exploitation of agent strategies let to study and reveal intrinsic game-theoretic properties of the logics for strategic reasonings, never grasped before. The most important one is that SL allows to specify sentences that can be satisfied only by agent strategies that are *not behavioral* [19, 21]. More specifically, in a determined history of a play, the value of a strategy may depend on what the other strategies will prescribe in the future or in other counterfactual plays. This means that, to choose an existential strategy, we need to know the entire structure of all universal strategies previously quantified. But this is in general unpredictable, as what we actually know is their value on the history of interest only. Clearly, using logics that admit non-behavioral strategies makes problematic their adoption in practical applications. Additionally, by allowing in SL such complicated strategies, we lose important model-theoretic properties and incur an increased complexity of related decision problems [19].

The quest for a behavioral semantics of SL has led to the definition of a settled family of syntactic restrictions [19]. Among the others, *Boolean-Goal Strategy Logic* (SL[bg], for short) encompasses sentences in a special prenex normal form having only a Boolean combination of temporal goals to handle at a time. For a goal, it is formally meant an SL formula of the form $\flat\psi$, where $\flat$ is a binding prefix of the type $(a_1, x_1), \ldots, (a_n, x_n)$ containing all the involved agents and ψ is a linear-time temporal logic formula, possibly expressed in LTL [26]. It has

been shown that SL[BG] admits non-behavioral strategies and, to avoid this, it is enough to limit the Boolean combination of goals just to a conjunction or a disjunction of them [21]. The corresponding logics, named *Conjunctive-Goal Strategy Logic* (SL[CG], for short) and *Disjunctive-Goal Strategy Logic* (SL[DG], for short), respectively, are the maximal syntactic fragment of SL known so far to admit a behavioral semantics and with a model-checking problem to be 2ExpTime-complete, as for ATL*. On the other hand, it is worth recalling that the exact complexity of the model-checking problem for SL[BG] is an open question and the best existing algorithm requires non-elementary time.

The positive results regarding SL[CG] and SL[DG] have left us with a conjecture in [21] that dealing with a fragment holding a behavioral semantics is a sufficient condition to ensure an elementary procedure for the related model-checking problem. This has stimulated us to introduce and study in this paper a whole hierarchy of syntactic behavioral fragments of SL[BG] and effectively show that the model-checking problem is elementary decidable. Precisely, we introduce *Alternating-Goal Strategy Logic* (SL[AG], for short) that, by imposing an opportune alternation over the nesting of Boolean connectives in SL[BG], induces two dual chains of behavioral classes of sentences, called *conjunctive-chain* and *disjunctive-chain*, of which SL[CG] and SL[DG] are just the base case. Analogously to the definition of classic dual hierarchies, each level i in a given chain is built recursively by making use of formulas at level $i-1$ of the other one. With more details, a matrix of a given level in these two chains has as form either $\phi \wedge \bigwedge_i \flat_i \psi_i$ or $\phi \vee \bigvee_i \flat_i \psi_i$, where ϕ belongs to the level $i-1$ of the dual chain. We grant the usefulness of SL[AG] by providing along the paper an example that requires a sentence belonging to the second level.

To give an idea of the shape that the internal matrix of an SL[AG] sentence can have, consider its parsing-tree along the Boolean connectives over goals. Such a tree has the property that, for each node, its labeling and the one of all its children but one coincide. This means that there is at most one path in the parsing-tree having an alternating interleaving of $\vee$ and $\wedge$ (this also provides an explanation for the name of the logic). Clearly, if such a path starts with $\wedge$, then the formula belongs to the conjunctive chain; otherwise, it belongs to the disjunctive one. Also, the length of the path determines the alternation level k of the class. Finally, observe that having in the parse-tree more than one node labeled differently from its father may already induce a sentence with a non-behavioral semantics [19, 21].

As a main result in this paper, we formally prove that classic and behavioral semantics for SL[AG] coincide. Additionally, we study the related model-checking problem showing that it is 2ExpTime-complete, thus not harder than that for ATL*. The latter result also keeps alive the conjecture mentioned above.

From a technical point of view, the specific restrictions imposed to SL[AG] formulas allow to simplify the reasoning about strategies by reducing this to a step-by-step analysis about which action to perform in each moment. With this observation in mind, we reduce the the satisfiability checking of a generic SL[AG] sentence over a given structure to that for a suitably *One-Goal Strategy*

Logic (SL[1G], for short) sentence over an ad hoc built structure. SL[1G] is, as expected from the name, a logic in which no Boolean combinations among goals are allowed [19]. This logic has the benefit of sharing with ATL* several positive structural properties, as well as, it results to be the maximal fragment of SL known so far to have a decidable satisfiability problem.

Due to space limit, most of the concepts related to SL and proofs are sketched. We refer to [19, 20, 23] for more material, motivations, and examples. Also, for recent works in strategic reasoning, one can see [1, 2, 4, 5, 7, 9, 25, 32].

2 Strategy Logic

In this section we introduce *Strategy Logic* [23]. Along the paper we use basic notation that, being standard, we omit and refer to [19] for a formal definition.

2.1 Game Structure

We start formalizing the game-theoretic framework on which the proposed strategic reasoning is performed. First, we introduce multi-agent concurrent arenas that, roughly speaking, describe the game board and its moves, *i.e.*, the physical world where agents act. Formally, an arena is defined as follows.

Definition 1. *Arena. - A multi-agent concurrent* arena *is a tuple* $\mathcal{A} \triangleq \langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \mathsf{tr} \rangle$*, where* Ag *is the finite set of* agents*,* a.k.a. players*,* Ac *is the set of* actions*,* a.k.a. moves*,* St *is the non-empty sets of* states*,* a.k.a. positions*. Assume* $\mathrm{Dc} \triangleq \mathrm{Ag} \rightharpoonup \mathrm{Ac}$ *to be the set of* decisions*,* i.e.*, partial functions describing the choices of an action by some agent. Then,* $\mathsf{tr} : \mathrm{Dc} \to (\mathrm{St} \rightharpoonup \mathrm{St})$ *denotes the* transition function *mapping every decision* $\delta \in \mathrm{Dc}$ *to a partial function* $\mathsf{tr}(\delta) \subseteq \mathrm{St} \times \mathrm{St}$ *representing a deterministic graph over the states.*

Informally, an arena can be seen as a generic *labeled transition graph*, where labels are agent decisions. However, in this work some conditions rule out how the transition function maps partial decisions to transitions. We preliminary introduce the set of decisions that trigger some transition in a given state $s \in \mathrm{St}$:

$$\mathsf{dc}(s) \triangleq \{\delta \in \mathrm{Dc} : s \in \mathsf{dom}(\mathsf{tr}(\delta))\};$$

As first property, we require is absence of end-states, *i.e.*, $\mathsf{dc}(s) \neq \emptyset$, for all $s \in \mathrm{St}$. Then, we need to provide a meaning to incomplete decisions. Roughly speaking, agents not mentioned in a decision are non influential, that is $(s_1, s_2) \in \mathsf{tr}(\delta)$ means that, in case the agents in $\mathsf{dom}(\delta)$ act as prescribed, the system goes from s_1 to s_2, no matter what the other agents do. This requires the following condition to be satisfied: for all $s \in \mathrm{St}$ and $\delta_1, \delta_2 \in \mathsf{dc}(s)$, there exists an agent $a \in \mathsf{dom}(\delta_1) \cap \mathsf{dom}(\delta_2)$ such that $\delta_1(a) \neq \delta_2(a)$. Finally, we assume that, each active agent in in a state $s \in \mathrm{St}$ is associated with a finite non-empty set of actions and all possible deriving combinations trigger some transition. First, the set of active agents in s and the relative associated actions are defined as follows:

$$\mathsf{ag}(s) \triangleq \{a \in \mathrm{Ag} : \exists \delta \in \mathsf{dc}(s) \,.\, a \in \mathsf{dom}(\delta)\},$$

$$\mathsf{ac}(s,a) \triangleq \{\delta(a) \in \mathrm{Ac} : \delta \in \mathsf{dc}(s) \ \wedge \ a \in \mathsf{dom}(\delta)\}.$$

Then, for all states s and decisions δ, if $\delta(a) \in \mathsf{ac}(s,a)$, for all $a \in \mathsf{ag}(s)$, we have that there is a decision $\delta' \in \mathsf{dc}(s)$ such that $\delta' \subseteq \delta$ (equivalently, $\delta{\restriction}_{\mathsf{dom}(\delta')} = \delta'$).

An arena $\mathcal{A}$ naturally induces a graph $\mathcal{G}(\mathcal{A}) \triangleq \langle \mathrm{St}, Ed \rangle$, where the edge relation $Ed \triangleq \bigcup_{\delta \in \mathrm{Dc}} \mathsf{tr}(\delta)$ is obtained by rubbing out all labels on the transitions. A path $\pi \in \mathrm{Pth}$ in $\mathcal{A}$ is simply a path in $\mathcal{G}(\mathcal{A})$. Similarly, the *order* $|\mathcal{A}| \triangleq |\mathcal{G}(\mathcal{A})|$ (resp., *size* $\|\mathcal{A}\| \triangleq \|\mathcal{G}(\mathcal{A})\|$) of $\mathcal{A}$ is the order (resp., size) of its induced graph. Finite paths also describe the possible evolutions of a play up to a certain point. For this reason, they are also called in the game-theoretic jargon *histories* and the corresponding set is denoted by $\mathrm{Hst} \triangleq \{\rho \in \mathrm{Pth} : |\rho| < \omega\}$.

A *strategy* is a function $\sigma \in \mathrm{Str} \triangleq \mathrm{Hst} \to \mathrm{Ac}$ prescribing which action has to be performed given a certain history. Roughly speaking, a strategy is a generic conditional plan which specifies *"what to do"* but not *"who will do it"*. We say that a strategy σ is *coherent* w.r.t. an agent a (a-coherent) if in each possible evolution of the game either a is not influential or the action that σ prescribes is available to a. Formally, for each history $\rho = s_0 \cdots s_n$, either $a \notin \mathsf{ag}(s_n)$ or $\sigma(\rho) \in \mathsf{ac}(s_n, a)$. A (strategy) *profile* $\xi \in \mathrm{Prf} \triangleq \mathrm{Ag} \to \mathrm{Str}$ specifies for each agent a coherent strategy. Given a profile ξ and an agent a, $\xi(a)(\rho)$ determines which action an agent a has chosen to perform on a history ρ. To identify, instead, the whole decision on ρ, we apply the flipping function $\widehat{\xi} : \mathrm{Hst} \to \mathrm{Dc}$.

A path π is *coherent w.r.t.* a profile ξ (ξ-*coherent*, for short) iff, for all $i \in [1, |\pi|[$, there exists a decision $\delta \in \mathsf{dc}(\pi_{i-1})$ such that $\delta \subseteq \widehat{\xi}(\pi_{<i})$ (equivalently, $\widehat{\xi}(\pi_{<i}){\restriction}_{\mathsf{dom}(\delta)} = \delta$) and $\pi_i = \mathsf{tr}(\delta)(\pi_{i-1})$, *i.e.*, π_i is the successor of π_{i-1} produced by the agent decision $\widehat{\xi}(\pi_{<i})$ prescribed by the profile ξ on the history $\pi_{<i}$. In case π is infinite, we say that it is a ξ-*play*. Note that, given a state s, the determinism of the arena ensures that there exists exactly one ξ-play π starting in s. Such a play is called (ξ, s)-*play* and is denoted by $\mathsf{play}(\xi, s)$.

As final remark, an arena is *turn-based* in case that for all states s, $|\mathsf{ag}(s)| \leq 1$.

An arena corresponds in the jargon of Modal Logics to a frame representing the "naked" structure of a model without any connection to the logic. Clearly, to check formulas, we need to interpret the atomic propositions over the states of the arena. We call a *concurrent game structure* the resulting structure.

Definition 2. *Concurrent Game Structure. - A* concurrent game structure *is a tuple* $\mathcal{G} \triangleq \langle \mathcal{A}, \mathrm{AP}, \mathsf{ap}, s_I \rangle$*, where* $\mathcal{A} \triangleq \langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \mathsf{tr} \rangle$ *is a multi agent concurrent arena,* AP *is finite non-empty sets of* atomic propositions*,* $s_I \in \mathrm{St}$ *is a designated* initial state*, and* $\mathsf{ap} : \mathrm{St} \to 2^{\mathrm{AP}}$ *is a* labeling function *that maps each state to the set of atomic propositions true in that state.*

As a running example, consider the arena $\mathcal{A}_S$ depicted in Figure 1. It represents a simple *scheduler system* in which two *processes*, $\mathtt{P}_1$ and $\mathtt{P}_2$, can require the access to a shared resource and an *arbiter* $\mathtt{A}$ is used to solve all conflicts that may arise. In particular, the arbiter can preempt a process owning the resource to allow the other one to access to it. The processes have three actions to interact with the system: $\mathtt{r}$ is used to request the resource from the system, when this is

not yet owned, while f releases it, when this is not necessary anymore and d is a "do-nothing" action in the case it does not want to change the present state. The arbiter, from its side, has two actions to decide which process has to receive the resource: 1 for $\mathtt{P}_1$ and 2 for $\mathtt{P}_2$.

The whole scheduler system can reside in the following four states: I, 1, 2, and A. The idle state I indicates that both processes do not own the resource, while i, with $i \in \{1, 2\}$, denotes that process $\mathtt{P}_i$ is using it. Finally, the arbitrage state A represents the situation in which an action from the arbiter is required in order to solve a conflict between contending requests. For readability reasons, a decision is graphically represented by an arrow $\mapsto$ with a sequence of agents on left hand side and the sequence of corresponding actions on right hand side. Finally, the arena is extended to a CGS by using I as initial state and atomic propositions c, a_1, and a_2 such that $\mathsf{ap}(\mathtt{A}) = \{c\}$, $\mathsf{ap}(1) = \{a_1\}$ and $\mathsf{ap}(2) = \{a_2\}$.

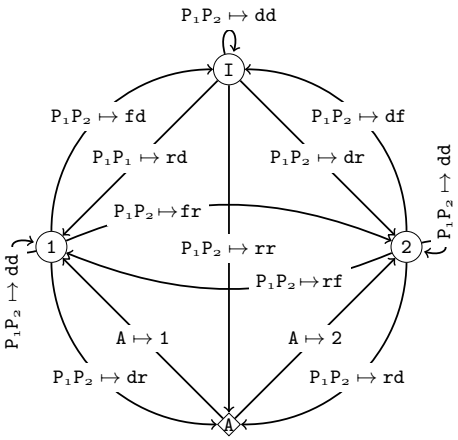

Fig. 1. Scheduler Arena $\mathcal{A}_S$.

2.2 Syntax

Strategy Logic extends LTL by introducing two *strategy quantifiers* $\langle\langle x \rangle\rangle$ and $[\![x]\!]$, and an *agent binding* (a, x). Informally, these operators can be respectively read as *"there exists a strategy x"*, *"for all strategies x"*, and *"bind agent a to the strategy associated with x"*. More formally, for each agent a we consider a countable set of dedicated variables Vr_a and with Vr we denote the union of all such variables. Then, SL well formed formulas are defined as follows.

Definition 3. *Syntax.* - SL formulas *are defined by the following grammar:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathtt{X}\varphi \mid \varphi\, \mathtt{U} \varphi \mid \varphi\, \mathtt{R} \varphi \mid \langle\langle x \rangle\rangle \varphi \mid [\![x]\!]\varphi \mid (a, x)\varphi,$$

where in (a, x) we assume that $x \in \mathrm{Vr}_a$.

The *free agents/variables* of a formula φ, $\mathsf{free}(\varphi)$, are the subset of $\mathrm{Ag} \cup \mathrm{Vr}$ 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 $[\![x]\!]$. A formula φ without free agents (resp., variables), i.e., with $\mathsf{free}(\varphi) \cap \mathrm{Ag} = \emptyset$ (resp., $\mathsf{free}(\varphi) \cap \mathrm{Vr} = \emptyset$), is named *agent-closed* (resp., *variable-closed*). If φ is both agent- and variable-closed, it is named *sentence*. By $\mathsf{snt}(\varphi)$ we denote the set of all sentences that are sub formulas of φ.

2.3 Semantics

Similarly as in FOL, the interpretation of a formula makes use of an assignment function which associates placeholders to some elements of the domain. In particular, an *assignment* is a (possibly partial) function $\chi \in \mathrm{Asg} \triangleq (\mathrm{Vr} \cup \mathrm{Ag}) \rightharpoonup \mathrm{Str}$ mapping variables and agents to strategies. An assignment χ is *complete* iff it is defined on all agents, *i.e.*, $\mathrm{Ag} \subseteq \mathsf{dom}(\chi)$. In this case, it directly identifies the profile $\chi_{\restriction \mathrm{Ag}}$ given by the restriction of χ to Ag. In addition, $\chi[e \mapsto \sigma]$, with $e \in \mathrm{Vr} \cup \mathrm{Ag}$ and $\sigma \in \mathrm{Str}$, is the assignment defined on $\mathsf{dom}(\chi[e \mapsto \sigma]) \triangleq \mathsf{dom}(\chi) \cup \{e\}$ which differs from χ only in the fact that e is associated with σ. Formally, $\chi[e \mapsto \sigma](e) = \sigma$ and $\chi[e \mapsto \sigma](e') = \chi(e')$, for all $e' \in \mathsf{dom}(\chi) \setminus \{e\}$.

The semantics of SL is defined as follows.

Definition 4. *Semantics. - Given a* CGS $\mathcal{G}$*, for all* SL *formulas* φ*, states* $s \in \mathrm{St}$*, and an assignment* $\chi \in \mathrm{Asg}$ *with* $\mathsf{free}(\varphi) \subseteq \mathsf{dom}(\chi)$*, the modeling relation* $\mathcal{G}, \chi, s \models \varphi$ *is inductively defined as follows.*

1. $\mathcal{G}, \chi, s \models p$ *if* $p \in \mathsf{ap}(s)$*, with* $p \in \mathrm{AP}$*.*
2. *Boolean operators are interpreted as usual.*
3. *For a variable* $x \in \mathrm{Vr}_a$*, it holds that:*
   (a) $\mathcal{G}, \chi, s \models \langle\!\langle x \rangle\!\rangle \varphi$ *if there exists an* a*-coherent strategy* $\sigma \in \mathrm{Str}$ *such that* $\mathcal{G}, \chi[x \mapsto \sigma], s \models \varphi$*;*
   (b) $\mathcal{G}, \chi, s \models [\![x]\!] \varphi$ *if for all* a*-coherent strategies* $\sigma \in \mathrm{Str}$ *it holds that* $\mathcal{G}, \chi[x \mapsto \sigma], s \models \varphi$*.*
4. $\mathcal{G}, \chi, s \models (a, x)\varphi$ *if* $\chi(x)$ *is coherent w.r.t.* a *and* $\mathcal{G}, \chi[a \mapsto \chi(x)], s \models \varphi$*.*
5. *Finally, if the assignment* χ *is also complete, for all formulas* φ*,* φ_1*, and* φ_2*, it holds that:*
   (a) $\mathcal{G}, \chi, s \models \mathtt{X}\varphi$ *if* $\mathsf{play}(\chi_{\restriction \mathrm{Ag}}, s) \models_{\mathrm{LTL}} \mathtt{X}\varphi$*;*
   (b) $\mathcal{G}, \chi, s \models \varphi_1 \mathtt{U} \varphi_2$ *if* $\mathsf{play}(\chi_{\restriction \mathrm{Ag}}, s) \models_{\mathrm{LTL}} \varphi_1 \mathtt{U} \varphi_2$*;*
   (c) $\mathcal{G}, \chi, s \models \varphi_1 \mathtt{R} \varphi_2$ *if,* $\mathsf{play}(\chi_{\restriction \mathrm{Ag}}, s) \models_{\mathrm{LTL}} \varphi_1 \mathtt{R} \varphi_2$*.*
   where $\models_{\mathrm{LTL}}$ *denotes the usual* LTL *semantics over paths.*

As the verification of a sentence φ does not depend on assignments, we omit them and write $\mathcal{G}, s \models \varphi$, for a generic s, and $\mathcal{G} \models \varphi$ when s is the initial state.

In the scheduler example, let $\varphi = \langle\!\langle x \rangle\!\rangle \langle\!\langle y_1 \rangle\!\rangle \langle\!\langle y_2 \rangle\!\rangle [\![z]\!] (\phi_1 \wedge \phi_2 \wedge \phi_3)$ where:

$$\phi_1 = (\mathtt{A}, x)(\mathtt{P}_1, y_1)(\mathtt{P}_2, z)\mathtt{G}(c \Rightarrow \mathtt{F}a_1),$$
$$\phi_2 = (\mathtt{A}, x)(\mathtt{P}_1, z)(\mathtt{P}_2, y_2)\mathtt{G}(c \Rightarrow \mathtt{F}a_2),$$
$$\phi_3 = [(\mathtt{A}, x)(\mathtt{P}_1, y_1)(\mathtt{P}_2, z)\mathtt{F}(c \Rightarrow \mathtt{X}a_1) \vee (\mathtt{A}, x)(\mathtt{P}_1, z)(\mathtt{P}_2, y_2)\mathtt{F}(c \Rightarrow \mathtt{X}a_2)]\,.$$

Informally, the formula φ expresses that, whenever a conflict arises, $\mathtt{A}$ has a strategy x to avoid that one of the processes jeopardizes the other one by preventing the latter to access the resource. Moreover, it requires that a processes will suddenly get the resource (*i.e.*, a step after the conflict arises).

It is easy to see that φ is satisfied in $\mathtt{I}$. Indeed, the arbiter strategy consists in alternating the access to the resource between the two processes, while they have to request it at most twice. Then, depending on an initial precedence, when the first conflict arises one of the two processes obtains the resource in the next state. Note that φ requires a unique strategy for the arbiter in order to coordinate with both processes independently. Therefore, it cannot be expressed in ATL*.

2.4 Fragments

Strategy Logic allows to freely compose LTL operators, bindings and strategy quantifiers. Such an expressiveness comes at a price of a NonElementaryTime complexity for the model checking problem. Therefore, it has been natural to investigate some syntactical fragments that can exhibit a better complexity.

A *quantification prefix* over a set $\mathrm{V} \subseteq \mathrm{Vr}$ is a finite word $\wp \in \{\langle\langle x \rangle\rangle, [\![x]\!] : x \in \mathrm{V}\}^{|\mathrm{V}|}$ of length $|\mathrm{V}|$ such that each variable $x \in \mathrm{V}$ occurs just once in $\wp$. By $\mathrm{Qn}(\mathrm{V})$ we indicate the set of quantification prefixes over V, whereas $\langle\langle \wp \rangle\rangle$ (resp. $[\![\wp]\!]$) denote the set of variables occurring *existentially* (resp. universally) quantified in $\wp$. Similarly, a *binding prefix* over V is a word $\flat \in \{(a, x) : a \in \mathrm{Ag} \wedge x \in \mathrm{V} \cap \mathrm{Vr}_a\}^{|\mathrm{Ag}|}$ such that each agent in Ag occurs exactly once in $\flat$. By Bn we indicate the set of all binding prefixes.

Definition 5. *Fragments.* - Boolean Goal SL *(*SL[BG] *for short) is defined by the following grammar:*

$$\varphi ::= \mathrm{LTL}(\varphi) \mid \wp\psi,$$

$$\psi ::= \flat\varphi \mid \neg\psi \mid \psi \wedge \psi \mid \psi \vee \psi,$$

where $\mathrm{LTL}(\varphi)$ *stands for the usual* LTL *grammar and* $\wp$ *quantifies over all free variables of* ψ*. The* One Goal SL *(*SL[1G]*),* Conjunctive Goal SL *(*SL[CG]*) and* Disjunctive Goal SL *(*SL[DG]*) are obtained from* SL[BG] *by restricting* ψ *to a single goal* $\flat\varphi$*, a conjunction of goals and a disjunction of goals, respectively.*

The relevance of the Boolean Goal fragment derives from the fact that the majority of strategic notions (e.g. Nash and Dominant equilibria) resides in this fragment. However, the precise complexity of the model checking problem is still an open issue, whereas SL[1G], SL[CG] and SL[DG] have been proved to be 2ExpTime-complete *w.r.t.* the length of the formula and PTime-complete *w.r.t.* the size of the model.

3 Behavioral Semantics

In this section we formalize the behavioral semantics. First of all, we provide an intuition of the concept of behavioralness with an example.

Consider the 2-agent turn-based CGS in Figure 2, where square states are ruled by agent α, while circle states by the opponent β. Note that each possible play consists in a sequence alternating circle and square states. Moreover, α and β are free to decide the truth value of p and q, respectively, in the next state.

Consider now the formula $\varphi' = \langle\langle x \rangle\rangle [\![y]\!] \langle\langle z \rangle\rangle ((\alpha, x)(\beta, y)\mathtt{F}q \longleftrightarrow (\alpha, z)(\beta, y)\mathtt{X}p)$. Clearly, this formula can be satisfied as follows: if the binding $(\alpha, x)(\beta, y)$ determines a path that eventually makes q true, then in the first step the strategy z has to choose the action 0, 1 otherwise. However, since β is free to decide whether and when q will be true, the agent α cannot respond step by step but has, at the beginning of the game, to guess about the future moves of β. In particular, we say that φ', even if satisfiable, is not behavioral.

Roughly, a formula is behavioral if it can be satisfied by assigning strategies that in all possible histories depend only on what the other strategies do on the same history. Therefore, behavioralness establishes a locality principle in the inter-dependence among strategies.

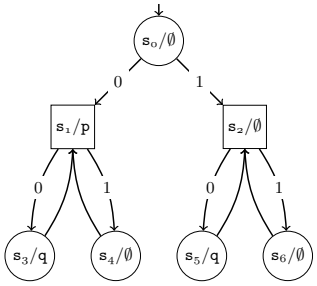

Fig. 2. A Critical Structure.

Formally speaking, we need to introduce the concept of *dependence map* which is analogous to the Skolemization procedure in first order logic. Then, we provide the notion of *elementariness*, a general functional correspondent of what behavioral means for strategies. Let $\wp \in \mathrm{Qn}(\mathrm{V})$ be a quantification prefix over a set $\mathrm{V} \subseteq \mathrm{Vr}$ of variables. For each variable $y \in \langle\!\langle \wp \rangle\!\rangle$, we use $\Delta(\wp, y)$ to denote the set of universally quantified variables $x \in [\![\wp]\!]$ 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 $\mathsf{v}\colon \mathrm{Vr} \rightharpoonup \mathrm{D}$. By $\mathsf{Val}_{\mathrm{D}}(\mathrm{V}) \triangleq \mathrm{V} \to \mathrm{D}$ we denote the set of all valuation functions over D whose domain is V.

A *dependence map* for $\wp$ over D is a function $\theta : \mathsf{Val}_{\mathrm{D}}([\![\wp]\!]) \to \mathsf{Val}_{\mathrm{D}}(\mathrm{V})$ satisfying the following properties: *(i)* $\theta(\mathsf{v})(x) = \mathsf{v}(x)$, for all $x \in [\![\wp]\!]$ and *(ii)*, for all $\mathsf{v}_1, \mathsf{v}_2 \in \mathsf{Val}_{\mathrm{D}}([\![\wp]\!])$ and $y \in \langle\!\langle \wp \rangle\!\rangle$, if $\mathsf{v}_{1\restriction\Delta(\wp,y)} = \mathsf{v}_{2\restriction\Delta(\wp,y)}$ then $\theta(\mathsf{v}_1)(y) = \theta(\mathsf{v}_2)(y)$, where $\mathsf{v}_{\restriction\Delta(\wp,y)}$ is the restriction of v to $\Delta(\wp, y)$. By $\mathsf{DM}_{\mathrm{D}}(\wp)$ 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 $\Delta(\wp,y)$.

Due to the fundamental Skolem theorem reported in [19], for each SL formula $\varphi = \wp\psi$ and CGS $\mathcal{G}$, we have that $\mathcal{G} \models \varphi$ iff there exists a dependence map $\theta \in \mathsf{DM}_{\mathrm{Str}}(\wp)$ such that $\mathcal{G}, \theta(\chi), s_{\mathrm{o}} \models \psi$, for all $\chi \in \mathrm{Asg}$ such that $[\![\wp]\!] \subseteq \mathsf{dom}(\chi)$. This substantially characterizes SL semantics by means of the concept of dependence map. Then, the behavioral semantics essentially constraints the set of dependence maps that can be used to satisfy a formula by requiring them to be *elementary*.

Elementariness is a purely functional notion defined through the concept of *adjoint function*. Let D, T, U, and V be four sets, and $\mathsf{m} : (\mathrm{T} \to \mathrm{D})^{\mathrm{U}} \to (\mathrm{T} \to \mathrm{D})^{\mathrm{V}}$ and $\widetilde{\mathsf{m}} : \mathrm{T} \to (\mathrm{D}^{\mathrm{U}} \to \mathrm{D}^{\mathrm{V}})$ two functions. Then, $\widetilde{\mathsf{m}}$ is the adjoint of m if $\widetilde{\mathsf{m}}(t)(\widehat{\mathsf{g}}(t))(x) = \mathsf{m}(\mathsf{g})(x)(t)$, for all $\mathsf{g} \in (\mathrm{T} \to \mathrm{D})^{\mathrm{U}}$, $x \in \mathrm{V}$, and $t \in \mathrm{T}$. Thus, a function m transforming a map of kind $(\mathrm{T} \to \mathrm{D})^{\mathrm{U}}$ into a new map of kind $(\mathrm{T} \to \mathrm{D})^{\mathrm{V}}$ has an adjoint $\widetilde{\mathsf{m}}$ if such a transformation can be done point wisely 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 6. *Elementary Dependence Maps. - Let* $\wp \in \mathrm{Qn}(\mathrm{V})$ *be a quantification prefix over a set* $\mathrm{V} \subseteq \mathrm{Vr}$ *of variables,* D *and* T *two sets, and* $\theta \in \mathsf{DM}_{\mathrm{T}\to\mathrm{D}}(\wp)$ *a dependence map for* $\wp$ *over* $\mathrm{T} \to \mathrm{D}$*. Then,* θ *is* elementary *if it admits an*

adjoint function. $\mathsf{EDM}_{\mathrm{T}\to\mathrm{D}}(\wp)$ *denotes the set of all elementary dependence maps for* $\wp$ *over* $\mathrm{T}\to\mathrm{D}$.

At this point, as mentioned above, we introduce a notion of *behavioral satisfiability*, in symbols $\models_{\mathrm{B}}$, which requires the elementariness of dependence maps over strategies.

Definition 7. *Behavioral Semantics. - Let* $\mathcal{G}$ *be a* CGS *and* $\varphi = \wp\psi$ *an* SL *sentence where* ψ *is agent-closed and* $\wp \in \mathrm{Qn}(\mathsf{free}(\psi))$. *Then,* $\mathcal{G}, s \models_{\mathrm{B}} \varphi$ *iff there exists a dependence map* $\theta \in \mathsf{EDM}_{\mathrm{Str}}(\wp)$ *such that* $\mathcal{G}, \theta(\chi), s \models \psi$, *for all* $\chi \in \mathrm{Asg}$ *such that* $[\![\wp]\!] \subseteq \mathsf{dom}(\chi)$.

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.

4 Alternating-Goal Strategy Logic

We now introduce *Alternating-Goal Strategy Logic* (SL[AG], for short), which we prove to have a behavioral semantics and an elementary model-checking problem.

4.1 Syntax

We start introducing the syntax of SL[AG], which extend both SL[CG] and SL[DG] by allowing to nest the Boolean connectives through a right-linear grammar.

Definition 8 (SL[AG] Syntax). *The syntax of* SL[AG] *is defined as follows:*

$$\varphi ::= \mathrm{LTL}(\varphi) \mid \wp\phi,$$
$$\phi ::= \flat\varphi \mid \flat\varphi \wedge \phi \mid \flat\varphi \vee \phi,$$

where $\wp \in \mathrm{Qn}(\mathsf{free}(\phi))$.

A sentence is *principal* if it is of the form $\wp\phi$, whereas it is *basic* in case the matrix ϕ generated by the second rule does not contain any further quantification. Also, with $\mathsf{bnd}(\phi)$, we mean the set of all bindings occurring in ϕ.

The introduced logic SL[AG] allows to identify two dual chains of fragments, called *conjunctive-chain* and *disjunctive-chain*, of which SL[CG] and SL[DG] are the base case. To give an intuition, by using an analogy to the definition of classic dual hierarchies, each level i in a chain is built recursively by making use of formulas at level $i-1$ of the other one. To make this concept formal, we first need to introduce some notions that will be also useful in the following.

In the following, by *alternating combination* over a given set of elements E, we simply mean a syntactic expression obtained by the grammar $\eta := e \mid e \wedge \eta \mid e \vee \eta$, where $e \in \mathrm{E}$. The set of all these combinations is denoted by $\mathtt{AC}(\mathrm{E})$. In addition, $\mathtt{AC}(\mathrm{E}, k) \subseteq \mathtt{AC}(\mathrm{E})$ indicates the subset of those combinations having the number of alternation between the connectives $\wedge$ and $\vee$ bounded by $k \in \mathbb{N}$. Finally, sup :

$\mathtt{BC}(\mathrm{E}) \to 2^{\mathrm{E}}$ is the function assigning to each combination $\eta \in \mathtt{BC}(\mathrm{E})$ its support $\mathsf{sup}(\eta) \subseteq \mathrm{E}$, *i.e.*, the set of elements on which it is built. Furthermore, by means of its overloading $\mathsf{sup} : \mathtt{BC}(\mathrm{E}) \times \mathbb{N} \to 2^{\mathrm{E}}$, we also denote the set $\mathsf{sup}(\eta, k) \subseteq \mathsf{sup}(\eta)$ of elements occurring in η at level $k \in \mathbb{N}$. As an example, consider the combination $\eta = \mathtt{e}_1 \wedge (\mathtt{e}_2 \vee \mathtt{e}_3 \vee (\mathtt{e}_4 \wedge \mathtt{e}_5)) \in \mathtt{AC}(\mathrm{E}, 3)$ over $\mathrm{E} = \{\mathtt{e}_i : i \in \mathbb{N}\}$. It is immediate to see that $\mathsf{sup}(\eta) = \{\mathtt{e}_1, \mathtt{e}_2, \mathtt{e}_3, \mathtt{e}_4, \mathtt{e}_5\}$ and $\mathsf{sup}(\eta, 2) = \{\mathtt{e}_2, \mathtt{e}_3\}$.

Observe that every matrix ϕ of an SL[AG] principal sentence $\wp\phi$ is an alternating combination over the set of goals $\{\flat\varphi : \flat \in \mathrm{Bn} \wedge \varphi \in \mathrm{SL[AG]}\}$. From this fact, we can easily derive the existence of a whole hierarchy of logics of which SL[CG] and SL[DG] are just the base case. Indeed, for each $k \in \mathbb{N}$, we can define the logics SL[k-CG] and SL[k-DG] as the fragments of SL[AG] obtained by only admitting matrices ϕ starting with a conjunction and a disjunction, respectively, and having alternation level bounded by k. Note that the sentence used as an example in Subsection 2.3 is a basic sentence of SL[2-CG].

4.2 Solution

We finally describe a polynomial reduction of the model-checking problem for SL[AG] to the same problem for SL[1G]. This reduction provides us with both a proof of the behavioral semantics and a decision procedure whose complexity is not higher than the one for ATL*, *i.e.*, 2ExpTime in the length of the specification and PTime in the size of the structure under analysis.

The reduction first consists of a conversion of the original CGS $\mathcal{G}$, on which we want to behaviorally verify the SL[AG] sentence $\varphi = \wp\phi$, in a new one $\mathcal{G}^\star$, in which we can keep track at the same time of a group of plays induced by φ, due to the different strategy bindings inside ϕ, that share a common history up to the current moment. Then, we check on the obtained structure a suitable SL[1G] sentence $\varphi^\star$ that, within its unique induced play, simulates those ones of the original sentence φ. In particular, every Boolean connective occurring in ϕ is replaced by a corresponding fresh agent in the new structure and their alternation is simply simulated by the one of the strategy quantifiers associated with these agents by means of the unique binding inside $\varphi^\star$.

To better describe the whole reduction, the transformation from $\mathcal{G}$ to $\mathcal{G}^\star$ is itself split in a conversion of the underlying arenas (see Construction 1) followed by a conversion of the associated labelings and initial states (see Construction 2).

The high-level idea behind the first construction is to build a composed arena $\mathcal{A}^\star$ in which each original state is paired with an alternating combination of bindings representing the part of the matrix ϕ of the original sentence $\varphi = \wp\phi$ to be still verified. The set of new agents is split into two components. First, we have the free variables of ϕ, called variable agents, that simulate the behavior of the original agents by choosing their actions between those of the original structure. Then, we have a fresh set of agents, one for each alternation level of the matrix, called numeric agents, that simulates the Boolean connective occurring in ϕ. Every one of the latter can either choose to verify a binding belonging to its own level or to pass the control to the successive agent. Finally, the new transition function just combines what the original one does for the binding chosen by

the last active numeric agent with the update of the current combination. In particular, the latter is obtained by restricting the combination to the set of all bindings that go together in the same determined direction.

Construction 1 (Arena Conversion) *From an arena* $\mathcal{A}=\langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \mathsf{tr}\rangle$ *and a* k*-bounded alternating combination* $\eta_I \in \mathtt{AC}(\mathrm{B}, k)$ *over a set of bindings* $\mathrm{B} \subseteq \mathrm{Bn}(\mathrm{Ag})$ *with* $k \in \mathbb{N}$*, we build the* composed arena $\mathcal{A}^\star \triangleq \langle \mathrm{Ag}^\star, \mathrm{Ac}^\star, \mathrm{St}^\star, \mathsf{tr}^\star\rangle$ *as follows:*

- *the new agents in* $\mathrm{Ag}^\star \triangleq \{x \in \mathrm{Vr} : \exists \flat \in \mathrm{B} \,.\, x \in \mathsf{rng}(\flat)\} \cup [1, k]$ *are represented by all variables bound by some binding and a range of numbers indicating all possible levels of alternation in* η_I*;*
- *the new actions in* $\mathrm{Ac}^\star \triangleq \mathrm{Ac} \cup \mathrm{B} \cup \{\oplus\}$ *are split into the original actions used by the variable agents and the bindings together with a fresh symbol used by the numeric agents;*
- *the new states in* $\mathrm{St}^\star \triangleq \mathrm{St} \times \mathtt{AC}(\mathrm{B}, k)$ *are pairs of original states and* k*-bounded alternating combinations over* B *indicating which parts of the initial combination* η_I *have to be verified from those states on.*

For each new state $s^\star = (s, \eta) \in \mathrm{St}^\star$*, we can describe its set of active new decisions: for all* $\delta^\star \in \mathrm{Dc}^\star$*, it holds that* $\delta^\star \in \mathsf{dc}^\star(s^\star)$ *iff there exists a number* $i \in [1, k]$ *such that*

- *the numeric agents from* 0 *to* i *are the only ones active on* $\delta^\star$*,* i.e.*,* $[1, i] \subseteq \mathsf{dom}(\delta^\star)$ *and* $]i, k] \cap \mathsf{dom}(\delta^\star) = \emptyset$*;*
- *all numeric agents up to* $i-1$ *decide to give the control on the bindings to the successive agent,* i.e.*,* $\delta^\star(j) = \oplus$*, for all* $j \in [1, i[$*;*
- *the numeric agent* i *chooses a binding occurring in the alternating combination* η *at level* i*,* i.e.*,* $\delta^\star(i) \in \mathsf{sup}(\eta, i)$*;*
- *the original decision* $\delta^\star \circ \delta^\star(i)$ *obtained by the composition of the actions chosen by the variable agents with the binding chosen by the numeric agent* i *is active on the original state* s*,* i.e.*,* $\delta^\star \circ \delta^\star(i) \in \mathsf{dc}(s)$*.*

To define the new transition function $\mathsf{tr}^\star$*, we first need to introduce the projection function* $\downarrow : \mathtt{AC}(\mathrm{B}, k) \times 2^{\mathrm{B}} \to \mathtt{AC}(\mathrm{B}, k)$ *that, given a combination* $\eta \in \mathtt{AC}(\mathrm{B}, k)$ *and a set of bindings* $\mathrm{B}' \subseteq \mathrm{B}$*, returns a new combination* $\eta{\downarrow}\mathrm{B}'$ *obtained by deleting from* η *all bindings not in* B'*. At this point, for each state* $s^\star = (s, \eta) \in \mathrm{St}^\star$ *and decision* $\delta^\star \in \mathsf{dc}^\star(s^\star)$ *with* $i \triangleq \mathsf{max}(\mathsf{dom}(\delta^\star) \cap [1, k])$*, we define the transition function* $\mathsf{tr}^\star(\delta^\star)(s^\star) \triangleq (s', \eta')$ *as follows:*

- *the original state* s' *is obtained as the successor of* s *following the original decision* $\delta^\star \circ \delta^\star(i)$ *obtained by functionally composing the new decision* $\delta^\star$ *with the function from* Ag *to* Vr *derived from the binding* $\delta^\star(i)$*,* i.e.*,* $s' \triangleq \mathsf{tr}(\delta^\star \circ \delta^\star(i))(s)$*;*
- *the combination* η' *is obtained from* η *by removing all bindings whose plays do not pass through the original state* s'*,* i.e.*,* $\eta' \triangleq \eta{\downarrow}\{\flat \in \mathsf{sup}(\eta) : s' = \mathsf{tr}(\delta^\star \circ \flat)(s)\}$*, where* $\delta^\star \circ \flat$ *is the original decision obtained by composing the new decision* $\delta^\star$ *with the function* $\flat : \mathrm{Ag} \to \mathrm{Vr}$ *derived from the binding.*

To complete the conversion of $\mathcal{G}$ into the composed $\mathcal{G}^\star$, we need to define both the initial state and the labeling of the latter structure. Obviously, the new initial state just ensures that all bindings are pointing to the original initial state, since the corresponding plays have to start synchronously. This is done by associating it with the original alternating combination. As the labeling concerns, to distinguish the bindings that are active on a given new state from those that are not, we further label it with the support of the associated combination.

Construction 2 (Structure Conversion) *From a* CGS $\mathcal{G} = \langle \mathcal{A}, \mathrm{AP}, \mathsf{ap}, s_I \rangle$ *and a* k*-bounded alternating combination* $\eta_I \in \mathtt{AC}(\mathrm{B}, k)$ *over a set of bindings* $\mathrm{B} \subseteq \mathrm{Bn}(\mathrm{Ag})$ *with* $k \in \mathbb{N}$*, we build the* composed CGS $\mathcal{G}^\star \triangleq \langle \mathcal{A}^\star, \mathrm{AP}^\star, \mathsf{ap}^\star, s_I^\star \rangle$ *as follows:*

- *the arena* $\mathcal{A}^\star$ *is built as in Construction 1;*
- *the new atomic propositions in* $\mathrm{AP}^\star \triangleq \mathrm{AP} \cup \mathrm{B}$ *are represented by the original atomic propositions augmented with the bindings;*
- *the new labeling function* $\mathsf{ap}^\star$ *assigns to each new state* $s^\star = (s, \eta) \in \mathrm{St}^\star$ *the set* $\mathsf{ap}^\star(s^\star) \triangleq \mathsf{ap}(s) \cup \mathsf{sup}(\eta)$ *of original atomic propositions holding in* s *together with the bindings in the support of the current combination* η*;*
- *the new initial state* $s_I^\star \triangleq (s_I, \eta_I)$ *is constituted by the original initial state extended with the combination* η_I*.*

Finally, we need to introduce the SL[1G] sentence $\varphi^\star = \wp^\star \flat^\star \psi^\star$ to verify on the composed structure $\mathcal{G}^\star$. Since this has to simulate the original SL[AG] sentence $\varphi = \wp\phi$, they have to share the same quantification prefix $\wp$. Moreover, we need to suitably quantify the strategies to associate with the numeric agents, in order to act in place of the Boolean connective inside ϕ. In the end, the LTL temporal goal $\psi^\star$ is directly obtained from ϕ by replacing all bindings occurring in it with an apposite check of its presence in the current play.

Construction 3 (Sentence Conversion) *From an* SL[k-CG] *(resp.,* SL[k-DG]*) basic sentence* $\varphi = \wp\phi$*, with* $k \in \mathbb{N}$*, we obtain the* SL[1G] *basic sentence* $\varphi^\star \triangleq \wp^\star \flat^\star \psi^\star$ *as follows:*

- *the new quantification prefix* $\wp^\star \triangleq \wp\wp'$ *is the extension of the original one* $\wp$ *with the strategy quantifications of the numeric agents whose type depends on the alternation level,* i.e.*,* $\wp' \triangleq \prod_{i=1}^{k} \mathtt{Qn}_i$*, with* $\mathtt{Qn}_i \triangleq \langle\langle i \rangle\rangle$*, if* $i \equiv 0 \pmod 2$ *(resp.,* $i \not\equiv 0 \pmod 2$*), and* $\mathtt{Qn}_i \triangleq [\![i]\!]$*, otherwise;*
- *the new injective binding* $\flat^\star \triangleq \prod_{x \in \mathsf{free}(\phi)} (x, x) \cdot \prod_{i=1}^{k} (i, i)$ *simply associates each variable quantified in* $\wp^\star$ *with the new agent having the same name;*
- *the* LTL *formula* $\psi^\star$ *is derived from* ϕ *by substituting each goal* $\flat\psi$ *with either* $\mathtt{G}\flat \wedge \psi$ *or* $\mathtt{G}\flat \rightarrow \psi$ *in dependence of the level in which it resides,* i.e.*,* $\psi^\star \triangleq \overline{\phi, 0}$*, where the translation function* $\overline{\cdot} : \mathrm{X} \times \mathbb{N} \rightarrow \mathrm{LTL}$ *is defined as follows:*
  - $\overline{\flat\psi, i} \triangleq \mathtt{G}\flat \wedge \psi$*, if* $i \equiv 0 \pmod 2$ *(resp.,* $i \not\equiv 0 \pmod 2$*), and* $\overline{\flat\psi, i} \triangleq \mathtt{G}\flat \rightarrow \psi$*, otherwise, where* $\flat \in \mathrm{B}$*,* $\psi \in \mathrm{LTL}$*, and* $i \in \mathbb{N}$*;*
  - $\overline{\mathtt{Cn}_h \phi_h, i} \triangleq \mathtt{Cn}_h \overline{\phi_h, i+1}$*, where* $\mathtt{Cn} \in \{\bigwedge, \bigvee\}$*,* $\phi_h \in \mathrm{X}$*, and* $i \in \mathbb{N}$*;*

  where X *is the set of all matrices of* SL[AG]*.*

We are now able to state the fundamental result about the reduction of the verification problem for SL[AG]. In the following, we shall then show how to use this reduction as a crucial building block on which to base the model-checking procedure for this fragment of SL.

Theorem 1 (SL[AG] Reduction). *Let $\mathcal{G}$ be a* CGS *and $\varphi = \wp\phi$ an* SL[AG] *basic sentence. Also, let $\mathcal{G}^\star$ be the composed* CGS *built in Construction 2, where $\eta_I \in \mathsf{AC}(\mathsf{bnd}(\phi))$ is the alternating combination over the set of bindings occurring in the matrix ϕ obtained by removing in the latter the* LTL *temporal part, and $\varphi^\star$ the* SL[1G] *basic sentence obtained in Construction 3. Then, $\mathcal{G} \models_{\mathrm{B}} \varphi$ iff $\mathcal{G}^\star \models \varphi^\star$.*

Proof. First observe that, once the *if* direction is proved, the *only if* direction immediately follows. Indeed, suppose by contradiction that $\mathcal{G} \models_{\mathrm{B}} \varphi$ but $\mathcal{G}^\star \not\models \varphi^\star$. Then, we have that $\mathcal{G}^\star \models \neg\varphi^\star$, so, $\mathcal{G}^\star \models (\neg\varphi)^\star$.[1] At this point, by the *if* direction, we have $\mathcal{G} \models_{\mathrm{B}} \neg\varphi$, but this is impossible.

To prove that $\mathcal{G}^\star \models \varphi^\star$ implies $\mathcal{G} \models_{\mathrm{B}} \varphi$, we simply show how to construct an elementary dependence map θ for the latter modeling relation starting from the one $\theta^\star$ of the former. Obviously, to do this we make use of the fact that SL[1G] is behavioral, *i.e.*, $\mathcal{G}^\star \models_{\mathrm{B}} \varphi^\star$.

As first thing, we need a partial function $\mathsf{ext} : \mathrm{Hst} \rightharpoonup \mathrm{Hst}^\star$, which maps each history in the original structure $\mathcal{G}$ used to verify the sentence φ to the corresponding one in the composed structure $\mathcal{G}^\star$, where the extension of the original states with the alternating combinations is done coherently with the decision chosen by the agents. Formally, we have that:

1. the original initial state s_I is mapped to the new initial state (s_I, η_I), *i.e.*, $s_I \in \mathsf{dom}(\mathsf{ext})$ and $\mathsf{ext}(s_I) \triangleq (s_I, \eta_I)$;
2. for each original history $\rho \in \mathsf{dom}(\mathsf{ext})$ already mapped to the new history $\rho^\star \triangleq \mathsf{ext}(\rho)$ having $s^\star = (s, \eta) \triangleq \mathsf{lst}(\rho^\star)$ as last state and for all new decisions $\delta^\star \in \mathsf{dc}^\star(s^\star)$ having $i \triangleq \mathsf{max}(\mathsf{dom}(\delta^\star) \cap \mathbb{N})$ as active numeric agent, it holds that $\rho \cdot s' \in \mathsf{dom}(\mathsf{ext})$ and $\mathsf{ext}(\rho \cdot s') \triangleq \rho^\star \cdot (s', \eta')$, where:
   - the original state s' is obtained as the successor of s following the original decision $\delta^\star \circ \delta^\star(i)$, *i.e.*, $s' \triangleq \mathsf{tr}(\delta^\star \circ \delta^\star(i))(s)$;
   - the combination η' is obtained from η by removing all bindings whose plays do not pass through the the original state s', *i.e.*, $\eta' \triangleq \eta{\downarrow}\{\flat \in \mathsf{sup}(\eta) : s' = \mathsf{tr}(\delta^\star \circ \flat)(s)\}$.

Now, we can easily define the elementary dependence map θ by means of the adjoint functions as follows: $\widehat{\theta}(\rho) \triangleq \widehat{\theta^\star}(\mathsf{ext}(\rho))$, for all $\rho \in \mathsf{dom}(\mathsf{ext})$. Observe that, we do not need to prescribe any constraint on the value $\widehat{\theta}(\rho)$, for $\rho \in \mathrm{Hst} \backslash \mathsf{dom}(\mathsf{ext})$, since these histories are not used in the verification of the sentence.

At this point, it just remains to prove that $\mathcal{G}, \theta(\chi), s_I \models \phi$, for all $\chi \in \mathrm{Asg}$ such that $\mathsf{dom}(\chi) = [\![\wp]\!]$. We leave this as an exercise.

By exploiting the above result, we can derive that classic and behavioral semantics for SL[AG] are equivalent, as stated in the following corollary.

[1] By $\neg\phi$, we actually mean the sentence in positive normal form equivalent to it.

Corollary 1 (SL[AG] Behavioral Semantics). *Let $\mathcal{G}$ be a* CGS *and φ an* SL[AG] *sentence. Then, $\mathcal{G} \models \varphi$ iff $\mathcal{G} \models_{\mathrm{B}} \varphi$.*

Proof. The proof simply proceeds by structural induction on the nesting of principal sentences. Here, we just show the base case, where $\varphi = \wp\phi$ is a basic sentence, and leave the easier inductive case to the reader. The *if* direction follows by definition of behavioral semantics. For the *only if* direction, we make use of a reasoning similar to the one done we have done in the previous theorem. Suppose by contradiction that $\mathcal{G} \models \varphi$ but $\mathcal{G} \not\models_{\mathrm{B}} \varphi$. Then, by Theorem 1, we have that $\mathcal{G}^\star \not\models \varphi^\star$, so, $\mathcal{G}^\star \models \neg\varphi^\star$, which in turn implies $\mathcal{G}^\star \models (\neg\varphi)^\star$. By using again the previous theorem, we have $\mathcal{G} \models_{\mathrm{B}} \neg\varphi$. Thus, by the *if* direction, we derive that $\mathcal{G} \models \neg\varphi$, which is impossible.

By inductively applying the reduction previously described on every principal subsentence of a given sentence of interest, we can reduce the model-checking problem of SL[AG] to a linear number of calls to the already known SL[1G] model-checking procedure [19, 22, 23]. Observe that, at each call, the arena conversion always applies to the original one. Instead, the structure conversion applies to a CGS augmented with a fresh proposition for each subsentence already analyzed. As for the CTL* model-checking procedure, the extra propositions only cost a linear factor in the computational complexity of the analyzed problem. From this, the following result directly derives.

Theorem 2 (Alternating Goal Complexity). *The model-checking problem of* SL[AG] *is 2*ExpTime-complete *in the length of the specification and* PTime-complete *in the size of the structure.*

Proof. As the lower bounds concern, the related results derive directly from the ATL* ones. Indeed, SL[AG] subsumes SL[1G], which in turn subsumes ATL*.

For the upper bound, consider a CGS $\mathcal{G}$ and an SL[AG] principal sentence $\varphi = \wp\phi$. If φ is basic, by exploiting the result of Theorem 1, we have that $\mathcal{G} \models \varphi$ iff $\mathcal{G}^\star \models \varphi^\star$. Therefore, the time complexity of the model-checking problem for φ against $\mathcal{G}$ is the sum of the time required by the reduction plus that of the same problem for $\varphi^\star$ against $\mathcal{G}^\star$. Both the time for building $\mathcal{G}^\star$ and its size are linear in the size of $\mathcal{G}$ and exponential in the number of bindings occurring in φ. The building of $\varphi^\star$, instead, is simply linear in the length of φ. The time complexity of the model checking for SL[1G], is known to be 2ExpTime-complete in the length of $\varphi^\star$ and PTime-complete in the size of $\mathcal{G}^\star$ [19]. Thus, the thesis for this case immediately follows.

If φ is not basic, we inductively solve the problem for all the immediate principal subsentences. Then, we enrich the labeling of $\mathcal{G}$ with fresh atomic propositions indicating whether a given subsentence holds in a certain state. Finally, we apply the procedure for the previous case on the new basic sentence φ' obtained from φ by substituting each immediate principal subsentence with the related atomic proposition. Hence, the thesis follows in this case too.

5 Discussion

In a recent paper titled "*What Makes* ATL* *Decidable? A Decidable Fragment of Strategy Logic*" [19], it has been argued for the first time that ATL* enjoys several positive properties thanks to its intrinsic *behavioral semantics*. In effect, several attempts of extending ATL* suffer from the fact that a choice of a strategy may depend on future strategies as well as on those over counterfactual plays. This is the case for both versions of *Strategy Logic* introduced in [6, 7] and [23].

As extensions of ATL* like SL are indispensable to represent key game-theoretic properties such as *Nash equilibria* and *sub-game perfect equilibria*[2], while the non-behavioral semantics is problematic to be implemented in practice, great effort has been devoted to find powerful fragments of SL for which the behavioral semantics suffices to evaluate the truth value of a formula.

Chronologically, the *Boolean-Goal* (SL[BG]) and the *One-Goal* (SL[1G]) fragments [19, 20], are the first two ones that have been investigated in this respect. It has been shown that, while SL[1G] enjoys all the main theoretic-properties of ATL*, including the behavioralness, the subsuming fragment SL[BG] does not. Notably, the model-checking problem for SL[1G] is 2ExpTime-complete (as for ATL*), while for SL[BG] the best existing algorithm requires non-elementary time. This has borne out the conjecture that dealing with a fragment holding a behavioral semantics is a sufficient condition to ensure an elementary procedure for such a decision problem. To enforce this, in [21], the *Conjunctive-Goal* (SL[CG]) and the *Disjunctive-Goal* (SL[DG]) fragments of SL[BG] have been introduced. Indeed, it has been shown that these logics strictly subsume SL[1G], are behavioral, and still retain a 2ExpTime-complete model-checking problem.

In this paper, we show that SL[CG] and SL[DG] are just at the bottom place of a hierarchy of behavioral fragments strictly contained in SL[BG] and that the conjecture still holds for all of them. Formally, we introduce Alternating-Goal Strategy Logic (SL[AG]) that imposes a precise alternation over the nesting of the Boolean connectives in SL[BG]. Precisely, in SL[AG], whenever there is a conjunction (disjunction), *at most one* of its conjunct (resp., disjunct) can be a disjunction (resp, conjunction). Note that allowing *two* instead of *one* conjunct/disjunct in the above definition would fall in a non-behavioral semantics [19].

As a future work, there are two lines of research we would like to follow. One is to prove the truth of the mentioned conjecture and possibly investigate whether it is also a *necessary* condition. The other one (also related to the necessary part of the conjecture) is to study the exact complexity of the model checking question of SL[BG], left open for some years now.

Acknowledgments

We thank an anonymous referee of the workshop LAMAS 2014 for having inspired us to introduce SL[AG].

[2] Note that sub-game perfect equilibria cannot be represented in the restricted turn-based two-player Strategy Logic version of Chatterjee, Henzinger and Piterman [10].

References

1. T. Ågotnes, V. Goranko, and W. Jamroga. Alternating-Time Temporal Logics with Irrevocable Strategies. In *TARK'07*, pages 15–24, 2007.
2. T. Ågotnes 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. 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.
5. P. Čermák, A. Lomuscio, F. Mogavero, and A. Murano. MCMAS-SLK: A Model Checker for the Verification of Strategy Logic Specifications. In *CAV'14*, LNCS 8559, pages 524–531. Springer, 2014.
6. K. Chatterjee, T.A. Henzinger, and N. Piterman. Strategy Logic. In *CONCUR'07*, LNCS 4703, pages 59–73. Springer, 2007.
7. K. Chatterjee, T.A. Henzinger, and N. Piterman. Strategy Logic. *IC*, 208(6):677–693, 2010.
8. E.A. Emerson and J.Y. Halpern. "Sometimes" and "Not Never" Revisited: On Branching Versus Linear Time. *JACM*, 33(1):151–178, 1986.
9. B. Finkbeiner and S. Schewe. Coordination Logic. In *CSL'10*, LNCS 6247, pages 305–319. Springer, 2010.
10. D. Fisman, O. Kupferman, and Y. Lustig. Rational Synthesis. In *TACAS'10*, LNCS 6015, pages 190–204. Springer, 2010.
11. W. Jamroga and A. Murano. On Module Checking and Strategies. In *AAMAS'14*, pages 701–708. International Foundation for Autonomous Agents and Multiagent Systems, 2014.
12. W. Jamroga and W. Penczek. Specification and Verification of Multi-Agent Systems. In *ESSLLI'11*, LNCS 7388, pages 210–263. Springer, 2011.
13. W. Jamroga and W. van der Hoek. Agents that Know How to Play. *FI*, 63(2-3):185–219, 2004.
14. O. Kupferman, M.Y. Vardi, and P. Wolper. Module Checking. *IC*, 164(2):322–344, 2001.
15. A. Lomuscio, H. Qu, and F. Raimondi. MCMAS: A Model Checker for the Verification of Multi-Agent Systems. In *CAV'09*, LNCS 5643, pages 682–688. Springer, 2009.
16. A. Lomuscio and F. Raimondi. MCMAS: A Model Checker for Multi-agent Systems. In *TACAS'06*, LNCS 3920, pages 450–454. Springer, 2006.
17. A. Lomuscio and F. Raimondi. Model Checking Knowledge, Strategies, and Games in Multi-Agent Systems. In *AAMAS'06*, pages 161–168. International Foundation for Autonomous Agents and Multiagent Systems, 2006.
18. A.D.C. Lopes, F. Laroussinie, and N. Markey. ATL with Strategy Contexts: Expressiveness and Model Checking. In *FSTTCS'10*, LIPIcs 8, pages 120–132. Leibniz-Zentrum fuer Informatik, 2010.
19. F. Mogavero, A. Murano, G. Perelli, and M.Y. Vardi. Reasoning About Strategies: On the Model-Checking Problem. Technical report, arXiv, 2011.
20. 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.

21. F. Mogavero, A. Murano, and L. Sauro. On the Boundary of Behavioral Strategies. In *LICS'13*, pages 263–272. IEEE Computer Society, 2013.
22. F. Mogavero, A. Murano, and L. Sauro. Strategy Games: A Renewed Framework. In *AAMAS'14*, pages 869–876. International Foundation for Autonomous Agents and Multiagent Systems, 2014.
23. F. Mogavero, A. Murano, and M.Y. Vardi. Reasoning About Strategies. In *FSTTCS'10*, LIPIcs 8, pages 133–144. Leibniz-Zentrum fuer Informatik, 2010.
24. 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.
25. S. Pinchinat. A Generic Constructive Solution for Concurrent Games with Expressive Constraints on Strategies. In *ATVA'07*, LNCS 4762, pages 253–267. Springer, 2007.
26. A. Pnueli. The Temporal Logic of Programs. In *FOCS'77*, pages 46–57. IEEE Computer Society, 1977.
27. L. Sauro and S. Villata. Dependency in Cooperative Boolean Games. *JLC*, 23(2):425–444, 2013.
28. S. Schewe. ATL* Satisfiability is 2ExpTime-Complete. In *ICALP'08*, LNCS 5126, pages 373–385. Springer, 2008.
29. P.Y. Schobbens. Alternating-Time Logic with Imperfect Recall. 85(2):82–93, 2004.
30. W. van der Hoek, W. Jamroga, and M. Wooldridge. A Logic for Strategic Reasoning. In *AAMAS'05*, pages 157–164. Association for Computing Machinery, 2005.
31. W. van der Hoek and M. Wooldridge. Cooperation, Knowledge, and Time: Alternating-Time Temporal Epistemic Logic and its Applications. *SL*, 75(1):125–157, 2003.
32. D. Walther, W. van der Hoek, and M. Wooldridge. Alternating-Time Temporal Logic with Explicit Strategies. In *TARK'07*, pages 269–278, 2007.