Strategy Games: A Renewed Framework

Fabio Mogavero
Università degli Studi di Napoli Federico II
fm@fabiomogavero.com

Aniello Murano
Università degli Studi di Napoli Federico II
murano@na.infn.it

Luigi Sauro
Università degli Studi di Napoli Federico II
luigi.sauro74@gmail.com

ABSTRACT

Game theory is a useful framework in multi-agent system verification. In this context, an important contribution is given by modal logics for strategic ability, such as ATL*, SL, and the like, which allow to describe the interaction among players for the achievement of specific goals. However, most of the attempts carried out so far have focused on specific forms of games in which strategic-reasoning aspects are glued with agent temporal goals.

In this paper, we propose a revisit of logics for strategic reasoning by following the general guidelines of game-theory, where the definition of solution concepts abstracts away the underlying cost-benefit analysis. Specifically, we introduce a game-theoretic framework consisting of three entities: the *arena*, shaping the rules of the game, the *extension* describing the property of interest of each possible play, and the *schema*, representing the agent roles and their interaction. In particular, to describe the latter, we use a variant of SL, by casting away its LTL temporal core. The new framework allows to gain in flexibility, modularity, and technical simplification, as well as, to grasp new features and better complexity results about logics previously studied. To give an evidence of this, by rephrasing SL in our framework, we improve known results about some of its fragments.

Categories and Subject Descriptors

I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Modal logic, Temporal logic, Representation languages*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*

General Terms

Theory, Verification

Keywords

Game models; Strategic logics; Formal verification.

1. INTRODUCTION

In the multi-agent system design and verification, logics for the strategic reasoning play a fundamental role [2,3,5,8,13,17].

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

These logics allow to check open system reliability by rephrasing the decision question as an interaction among different players in a game-strategic setting, which may compete or collaborate for the achievement of specific *goals*. A pioneering logic in this matter is *Alternating-Time Temporal Logic* (ATL*, for short) [2]. This logic makes use of strategic modalities of the form "$\langle\langle \mathrm{A} \rangle\rangle$" and "$[[\mathrm{A}]]$", for a set A of *agents*. Their formulas are interpreted over *concurrent game structures* (CGS, for short) [2], which are transition systems modeling the interaction among agents (a.k.a. *players*) and whose states are labeled with a set of atomic proposition true in that state. Given a CGS $\mathcal{G}$ and a set A of agents, an ATL* formula $\langle\langle \mathrm{A} \rangle\rangle\psi$ is satisfied at a state s of $\mathcal{G}$ if there is a set of strategies for agents in A such that, no matter which strategies are executed by agents not in A, the resulting outcome of the interaction in $\mathcal{G}$ satisfies the temporal goal ψ at s. In particular, ψ is build upon the syntax rules of the classical linear-time temporal-logic LTL [15]. Since its introduction, ATL* has been intensively studied and its model checking has been proved to be 2ExpTime-complete [2]. Also, several practical applications of this logic have been investigated through the implementation of useful verification tools [6].

The benefits of ATL* brought to multi-agent system verification have opened to a broader modeling scenario and several variants of this logic have been successively introduced and studied, in order to balance expressiveness with respect to complexity of the related decision questions [1,3,7,10,11,13,19]. Among the others, a recent story of success is *Strategy Logic* (SL, for short) [13]. This logic has the benefit to include all main logics for strategic reasoning previously introduced and to be powerful enough to express fundamental concepts of games, such as *Nash equilibrium*, which instead cannot be expressed in ATL*, unless one considers substantial extensions [16].

A key aspect of SL is that, differently from ATL*, strategies are not treated implicitly through its modalities, but rather used as 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. In SL, 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. Then, to use a strategy, an explicit *binding* operator (a, x) allows to bind an agent a to the strategy associated with a variable x. On the other hand, SL uses CGSs as underling structures as well as temporal goals are given in terms of LTL rules, as it is for ATL*.

Although ATL* and SL count several interesting applications in computer science, they only use a restricted power offered by game theory. This is mainly due to the rigidity of the syntax used to define these logics, which overall come out as an evolution of model checking and temporal logics, originally conceived for the verification of closed systems. Indeed, from one side we use CGS as a model, which is a Kripke structure further equipped with a description of player moves. Hence, we inherit an intrinsic labeling of the states with atomic propositions. From the other side, we extend path modalities with game modalities, but we limit ourself to use predicates over paths, *i.e.*, game evolutions, mainly expressed in terms of LTL formulas. Additionally, this part is glued inside the syntax of the logic itself. In game theory speaking, player rules, goals, and verification properties are independent entities, usually given and treated separately. Conversely, having them all glued in one object leads to the loss of several important features, as well as, it limits the applicability of the resulting model. Yet in terms of game theory, combining these entities is as bizarre as letting a solution concept to depend from the specific form of agents utility functions. Consider for example Nash equilibria. Using a somewhat standard notation, a strategy profile ξ is a Nash equilibrium if, for each other strategy profile ξ' consisting in a unilateral deviation of an agent i from ξ, $\mathsf{u}_i(\xi') \leq \mathsf{u}_i(\xi)$, where u_i is the utility function of the agent i. Notice that, since the utility functions occur as generic functions, the definition of Nash equilibria does not rely on how they are determined: different cost-benefit analysis methods can be used as a black box in the pure strategic reasoning.

In this paper, we propose a revisit of classical logics for strategic reasoning by following the general guidelines of the game-theory where the definition of solution concepts (*i.e.*, the pure strategic reasoning) abstracts away the underlying cost-benefit analysis (together with the temporal constraints used to determinate it). Specifically, we introduce a game-theoretic framework consisting of three entities: the *arena*, shaping the rules of the game, the *extension* describing the property of interest of each possible play, and the *schema*, representing properties of agent interactions (*e.g.* Nash equilibrium). In particular, to describe the latter, we use a variant of SL, by casting away its LTL temporal core.

We show that the introduced framework has several advantages. First, it allows to reason about the game model in a modular way. More specifically, we can investigate the complexity of a decision problem by looking at the exact contribution of each component of the strategic reasoning. As often it happens in multi-agent system verification, system and specification parameters may vary significantly one from another and the best methodology to be used to address a specific decision problem is a trade-off. Second, one can embed different formalisms to describe temporal goals. In particular, we can use very expressive extensions of the LTL syntax rules by using the linear μCALCULUS [18], or vice versa use a very restricted form of LTL such as the common fragment of LTL and CTL. As another example, one can inject goals directly represented by means of automata [20].

As a positive side effect, the proposed game framework let us to re-address the main features of SL and obtain new and better complexity results regarding the model-checking problem of some of its fragments, once the related extension is represented by means of classic LTL formulas. Specifically, three fragments of SL have been recently introduced and deeply investigated: SL[1G], SL[CG], and SL[DG] [10–12]. The latter two logics, which include the former, are at the moment the maximal fragments of SL known to have an elementary model-checking and precisely 2EXPTIME-COMPLETE. Here, we give new game-based decision procedures for the model-checking problem of all these fragments, which allow to improve their overall complexities. Notably, we use as decision tool of the game framework, the framework itself.

Due to space limit, most of the proofs are just sketched and reported in the full version of the article.

2. STRATEGY GAMES

In this section, we introduce a reformulation of the game-theoretic framework through which we model strategic reasonings. Differently from other approaches, where a unique structure is defined, here we separate the basic aspects of a game in three components: arena, extension, and schema. Roughly speaking, an arena describes the game board and its moves, *i.e.*, the physical world where agents act. An extension specifies object-level conditions over game evolutions, *i.e.*, how to determine the winner of a play. Finally, a schema abstractly represents strategic properties of the game, like determinacy, existence of Nash equilibria, and so on.

2.1 Arenas

We now formally define the concept of arena.

DEFINITION 2.1. *Arena. - A multi-agent concurrent* arena *is a tuple* $\mathcal{A} \triangleq \langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \mathsf{tr} \rangle \in \mathrm{Ar}(\mathrm{Ag})$*, 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*, and* $\mathrm{Ar}(\mathrm{Ag})$ *is the class of all arenas over* Ag*. 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,* $\mathsf{tr} : \mathrm{Dc} \rightarrow (\mathrm{St} \rightharpoonup \mathrm{St})$ *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.*

Intuitively, an arena can be seen as a *labeled transition graph*, where labels are possibly incomplete agent decisions, which determine the transition to be executed at each step of a play in dependence of the choices made by the agents in the relative state. Incomplete decisions allow us to represent any kind of legal move in a state, where some agents or a particular combination of actions may not be active. Note that, since $\mathsf{tr}(\delta)$ is a partial function, we are implicitly assuming that the arena is deterministic, *i.e.*, a decision $\delta \in \mathrm{Dc}$ may be not applicable in a state $s \in \mathrm{St}$, but if it is, then it leads to the unique successor $\mathsf{tr}(\delta)(s)$.

An arena $\mathcal{A}$ naturally induces a graph $\mathcal{G}(\mathcal{A}) \triangleq \langle \mathrm{St}, Ed \rangle$, whose vertexes are represented by the states and 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. Note that there could be states where no transitions are available, *i.e.*, $\mathsf{dom}(Ed) \subset \mathrm{St}$. If this is the case, the states in $\mathrm{St} \setminus \mathsf{dom}(Ed)$ are called *sink-states*.

As usual in the study of extensive-form games, finite paths also describe the possible evolutions of a play up to a certain point. For this reason, they are 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 partial function $\sigma \in \mathrm{Str} \triangleq \mathrm{Hst} \rightharpoonup \mathrm{Ac}$ prescribing, whenever defined, which action has to be performed for a certain history of the current outcome. In addition, a *profile* is a function $\xi \in \mathrm{Prf} \triangleq \mathrm{Ag} \to \mathrm{Str}$ assigning to every agent the adopted strategy. Roughly speaking, a strategy is a generic conditional plan to achieve one or more goals, which specifies *"what to do"* but not *"who will do it"*, while a profile specifies a unique behavior for each agent by associating her with a strategy. Note that a given strategy can be used by more than one agent at the same time.

For a profile $\xi \in \mathrm{Prf}$, to identify which action an agent $a \in \mathrm{Ag}$ has chosen to perform on a history $\rho \in \mathrm{Hst}$, we first extract the corresponding strategy $\xi(a)$ and then we determinate the action $\xi(a)(\rho)$, whenever defined. To identify, instead, the whole decision on ρ, we apply the flipping operator to ξ. We get so a function $\widehat{\xi} : \mathrm{Hst} \to \mathrm{Dc}$ such that $\widehat{\xi}(\rho)(a) = \xi(a)(\rho)$, which maps each history to the planned decision.

A path $\pi \in \mathrm{Pth}$ is *coherent w.r.t.* a profile $\xi \in \mathrm{Prf}$ (ξ-*coherent*, for short) iff, for all $i \in [1, |\pi|[$, it holds that $(\pi)_i = \mathsf{tr}(\widehat{\xi}((\pi)_{<i}))((\pi)_{i-1})$, *i.e.*, $(\pi)_i$ is the unique successor of $(\pi)_{i-1}$ identified by the agent decision $\widehat{\xi}((\pi)_{<i})$ described in the profile ξ on the history $(\pi)_{<i}$. Moreover, π is a ξ-*play* iff it is a maximal ξ-coherent path, *i.e.*, it is not a proper prefix of any another ξ-coherent path. Clearly, an infinite path is ξ-coherent iff it is a ξ-play. Given a state $s \in \mathrm{St}$, the determinism of the arena ensures that there exists exactly one ξ-play π starting in s, *i.e.*, $\mathsf{fst}(\pi) = s$. Such a play is called (ξ, s)-*play*. For this reason, we use the *play function* $\mathsf{play} : \mathrm{Prf} \times \mathrm{St} \to \mathrm{Pth}$ to identify, for each profile $\xi \in \mathrm{Prf}$ and state $s \in \mathrm{St}$, the corresponding (ξ, s)-*play* $\mathsf{play}(\xi, s)$. Observe that finite ξ-plays do not necessarily end in a sink-state, since it may be that it cannot be extended in a way that is coherent *w.r.t.* the profile ξ (below we return on this point).

Arenas describe generic mathematical structures, where the basilar game-theoretic notions of history, strategy, profile, and play can be defined. However, in several contexts, some constraints rule out how the function tr maps partial decisions to transitions. For example, some definitions of *bounded* concurrent arenas require that, in a given state, each agent is associated with a finite non-empty set of actions, whose possible combinations are all applicable in that state. Another case is that of classic *turn-based* arenas, where there is at most one active agent for each state. Both these examples represent particular cases of our general definition of arena, since they can be embedded in this formalism by forcing the transition function to satisfy some further condition. In order to illustrate such embeddings, by means of the functions $\mathsf{dc} : \mathrm{St} \to 2^{\mathrm{Dc}}$ and $\mathsf{ag} : \mathrm{St} \to 2^{\mathrm{Ag}}$ defined as follows, we preliminary introduce the set of decisions and agents that trigger some transition in a given state $s \in \mathrm{St}$:

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

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

Thanks to this notation, a restriction to bounded concurrent arenas can be provided by saying that, for each state $s \in \mathrm{St}$, there is a left-total relation $R \subseteq \mathrm{Ag} \times \mathrm{Ac}$ with $|\mathsf{rng}(R)| < \omega$ such that $\mathsf{dc}(s) = \{\delta \in \mathrm{Dc} : \delta \subseteq R\}$. Similarly, a restriction to turn-based arenas is given by $|\mathsf{ag}(s)| \leq 1$.

As a running example, used to illustrate the introduced definitions, consider the arena $\mathcal{A}_S$ depicted in Figure 1. It represents a model of a simple *scheduler system* in which two *processes*, $\mathtt{P}_1$ and $\mathtt{P}_2$, can require the access to a *shared resource*, like a processor, and an *arbiter* $\mathtt{A}$ is used to solve all conflicts that may arise when contending requests are made. In particular, the arbiter can preempt a process owning the resource to allow the other one to access to it. The processes have two actions to interact with the system: $\mathtt{r}$ for request and $\mathtt{f}$ for free/release. The former is used to request the resource from the system, when this is not yet owned, while the latter releases it, when this is not needed anymore. In addition, a process can simply avoid to make any action, in case it wants to remain in the current state. The arbiter, from its side, has two actions to decide which process has to receive the resource: $\mathtt{1}$ for $\mathtt{P}_1$ and $\mathtt{2}$ for $\mathtt{P}_2$. The whole scheduler system can reside in the following four states: $\mathtt{I}$, $\mathtt{1}$, $\mathtt{2}$, and $\mathtt{A}$. The idle state $\mathtt{I}$ indicates that both processes do not own the resource, while i, with $i \in \{\mathtt{1}, \mathtt{2}\}$, that process $\mathtt{P}_i$ is using it. Finally, the arbitrage state $\mathtt{A}$ 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. Formally, $\mathcal{A}_S$ is the arena $\langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \mathsf{tr} \rangle$, where all components but the transition function are set as follows: $\mathrm{Ag} = \{\mathtt{A}, \mathtt{P}_1, \mathtt{P}_2\}$, $\mathrm{Ac} = \{\mathtt{r}, \mathtt{f}, \mathtt{1}, \mathtt{2}\}$, and $\mathrm{St} = \{\mathtt{I}, \mathtt{1}, \mathtt{2}, \mathtt{A}\}$. In addition, we have that $\mathsf{ag}(s) = \{\mathtt{P}_1, \mathtt{P}_2\}$, for all $s \in \{\mathtt{I}, \mathtt{1}, \mathtt{2}\}$, and $\mathsf{ag}(\mathtt{A}) = \{\mathtt{A}\}$. Moreover, $\mathsf{dc}(\mathtt{I}) = \{\varnothing, \mathtt{P}_1 \mapsto \mathtt{r}, \mathtt{P}_2 \mapsto \mathtt{r}, \mathtt{P}_1\mathtt{P}_2 \mapsto \mathtt{rr}\}$, $\mathsf{dc}(\mathtt{A}) = \{\mathtt{A} \mapsto \mathtt{1}, \mathtt{A} \mapsto \mathtt{2}\}$, and $\mathsf{dc}(i) = \{\varnothing, \mathtt{P}_i \mapsto \mathtt{f}, \mathtt{P}_{3-i} \mapsto \mathtt{r}, \mathtt{P}_i\mathtt{P}_{3-i} \mapsto \mathtt{fr}\}$, for all $i \in \{1, 2\}$.

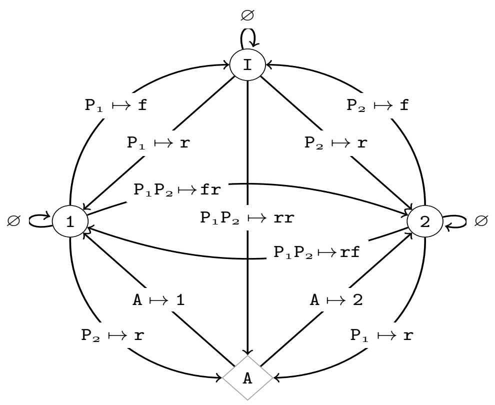


Figure 1: Scheduler Arena $\mathcal{A}_S$.

2.2 Extensions

Clearly, a game, besides its dynamics, consists also of private objectives or a global goal, which typically require to check some properties over paths. For instance, relevant conditions in chess are whether the game ends up in a checkmate or a stalemate, which determine a winner or a draw, respectively. More in general, path properties may represent fairness, reachability, safety, or other conditions, usually expressed by means of some kind of temporal language, like LTL [15] or the linear μCALCULUS [18], which combine atomic propositions over states with suitable temporal operators. As game theorists usually do, our aim is to separate the strategic reasoning from the specification of such properties and represent them as generic Boolean conditions over paths.

DEFINITION 2.2. *Extension.* - *An* extension *is a tuple* $\mathcal{E} \triangleq \langle \mathcal{A}, \mathrm{Pr}, \mathsf{pr} \rangle \in \mathrm{Ex}(\mathrm{Ag}, \mathrm{Pr})$*, where* $\mathcal{A} \in \mathrm{Ar}(\mathrm{Ag})$ *is the underlying arena,* Pr *is the finite non-empty set of* predicates*,* $\mathsf{pr} : \mathrm{Pr} \to 2^{\mathrm{Pth}}$ *is the* predicate function *mapping each* *predicate* $p \in \mathrm{Pr}$ *to the set of paths* $\mathsf{pr}(p)$ *satisfying it, and* $\mathrm{Ex}(\mathrm{Ag}, \mathrm{Pr})$ *is the class of all extensions over* Ag *and* Pr.

Intuitively, an extension classifies paths by means of a set of monadic predicates that describe relevant temporal properties of plays for the specific domain under analysis.

An extension is *Borelian* (resp., *regular*) iff, for all predicates $p \in \mathrm{Pr}$, the induced language $\mathsf{pr}(p) \subseteq \mathrm{St}^{\infty}$ of finite and infinite words over the states is Borelian (resp., regular) [14].

Continuing with the scheduler example, consider the regular extension $\mathcal{E}_S$ consisting of the scheduler arena $\mathcal{A}_S$ and two predicates p_1 and p_2 expressing that, whenever a conflict arises, every process $\mathtt{P}_k$ will have at some point in the future access to the resource. Formally, for each path $\pi \in \mathrm{Pth}$, it holds that $\pi \in \mathsf{pr}(p_k)$ iff, for all $i \in [0, |\pi|[$ with $(\pi)_i = \mathtt{A}$, there exists $j \in]i, |\pi|[$ such that $(\pi)_j = \mathtt{k}$.

2.3 Schemas

Coming back to the chess analysis, note that we have defined the chessboard together with the possible legal moves (the arena). Also, we have introduced the relevant properties, such as checkmate or stalemate, which have to be checked during the game (the extension). However, we have not yet completely grasped the notion of game. Indeed, an initial configuration needs to be specified and, more important, it is necessary to indicate which behavior we expect from the players. To do this, we introduce the concept of schema. It abstractly describes the possible roles and the allowed interactions of agents, by means of a relation between an extension, with an associated initial state, and some elements describing which behaviors that extension satisfies.

Definition 2.3. *Schema. - A* schema *over sets of agents* Ag *and predicates* Pr *is a tuple* $\mathcal{S} \triangleq \langle \mathrm{Tr}, \models \rangle \in \mathrm{Sc}(\mathrm{Ag}, \mathrm{Pr})$, *where* Tr *is the non-empty set of* targets, $\models \subseteq \mathrm{Ex}(\mathrm{Ag}, \mathrm{Pr}) \times_{\mathcal{E}} \mathrm{St}_{\mathcal{E}} \times \mathrm{Tr}$ *is the* schema relation *describing which targets* $t \in \mathrm{Tr}$ *can be achieved on an extension* $\mathcal{E} \in \mathrm{Ex}(\mathrm{Ag}, \mathrm{Pr})$ *starting from a given initial state* $s \in \mathrm{St}_{\mathcal{E}}$*, in symbols* $\mathcal{E}, s \models t$*, and* $\mathrm{Sc}(\mathrm{Ag}, \mathrm{Pr})$ *is the class of all schemas over* Ag *and* Pr.

In the scheduler example, an interesting target t is to avoid starvation for a process, due to the selfish behavior of the other one. To precisely state t, one can either define an ad-hoc schema relation or introduce a suitable formal language that allows to describe an entire class of possible schemas. Clearly, we follow the second approach from now on.

2.4 Games

By summing up the definitions of arena, extension, and schema, we now formalize the concept of game.

Definition 2.4. *Game. - Let* $\mathcal{S} \in \mathrm{Sc}(\mathrm{Ag}, \mathrm{Pr})$ *be a schema over the sets of agents* Ag *and predicates* Pr*. Then, a* game w.r.t. $\mathcal{S}$ *is a tuple* $\partial \triangleq \langle \mathcal{E}, s, t \rangle \in \mathrm{Gm}(\mathcal{S})$*, where* $\mathcal{E} \in \mathrm{Ex}(\mathrm{Ag}, \mathrm{Pr})$ *is the underlying extension,* $s \in \mathrm{St}_{\mathcal{E}}$ *is the designated initial state,* $t \in \mathrm{Tr}$ *is the prescribed target, and* $\mathrm{Gm}(\mathcal{S})$ *is the class of all games over* $\mathcal{S}$.

A game $\partial \triangleq \langle \mathcal{E}, s, t \rangle$ is *fulfilled* iff $\mathcal{E}, s \models t$. Intuitively, this means that all agents can achieve the target t on the predicate arena $\mathcal{E}$ starting from s. The *fulfillment problem* is to decide whether a game is fulfilled. Observe that this problem is an abstract generalization of the model-checking problem of a language against a model, which we here respectively represent via a schema and an extension.

3. STRATEGY LOGICS

By analogies with object-oriented languages, a schema can be seen as an interface which allows to describe the notion of fulfillment of a game in a very general way. Following the same analogy, a schema has to be implemented in order to make the game concrete. It is for this reason that we have introduced schemas in our framework. Indeed one can change an arena or an extension without changing the schema. Clearly, the more powerful the language used to formalize the schema is the wider the class of strategic notions we can grasp is as well. Hence, we borrow the strategic modalities from SL, instead of the less powerful coalition modalities of ATL*, as a language for the targets, and its semantics as the corresponding schema relation.

3.1 Syntax

Strategic reasoning primary concerns to analyze different counterfactual hypothesis such as *"what if two agents cooperate..."*, *"what if another agent tries to obstruct them..."*, or *"what if an agent deviates from a given behavior..."*. From a logical point of view, this means to consider different patterns of quantification over strategies, as well as, to bind such strategies to the agents. For this reason, we introduce two *strategy quantifiers*, the existential $\langle\langle x \rangle\rangle$ and the universal $[[x]]$, and an *agent binding* (a, x), where a is an agent and x belongs to a countable set Vr of variables. Intuitively, 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"*. Clearly, such a bag of logical quantifiers would be completely pointless without the possibility to express properties over paths. Hence, the set Pr of predicates is part of our syntax together with standard Boolean operators.

It remains to discuss an important issue regarding which strategies can be assigned to the agents. As mentioned in the previous section, given a profile ξ and an initial state s, the play $\pi = \mathsf{play}(\xi, s)$ can be a finite path even if its last state $\mathsf{lst}(\pi)$ is not a sink-state. Formally, this happens when the decision $\widehat{\xi}(\pi)$ is not in $\mathsf{dc}(\mathsf{lst}(\pi))$ even if $\mathsf{dc}(\mathsf{lst}(\pi)) \neq \emptyset$. To give an intuitive example, a chess match could be trivially blocked in its initial configuration just because the white player decides not to perform any move. Clearly, it is a matter of the specific application domain whether the agents are free to adopt any strategy, even the blocking ones, or they are forced to trigger a transition whenever it is possible. To be as general as possible, we do not implicitly restrict the strategies that can be assigned to the agents, but we introduce a further constant $\odot$ to express the fact that we are restricting to no blocking assignments.

Definition 3.1. *Syntax.* - SL formulas *are defined by means of the following context-free grammar, where* $a \in \mathrm{Ag}$, $p \in \mathrm{Pr}$, *and* $x \in \mathrm{Vr}$:

$$\varphi := p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle\langle x \rangle\rangle\varphi \mid [[x]]\varphi \mid (a, x)\varphi \mid \odot.$$

Usually, to provide the semantics of a predicative logic, it is necessary to define the concepts of free and bound *placeholders*. For example, first order logic has the variables as unique type of placeholder. In SL, instead, since strategies can be associated to both agents and variables, we use the set of *free agents/variables* $\mathsf{free}(\varphi)$ as 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 predicate p and *(ii)* all variables x for which there is a binding (a, x) but no quantification $\langle\langle x \rangle\rangle$

or $[\![x]\!]$. In case $\mathsf{free}(\varphi) \cap \mathrm{Ag} = \emptyset$ (resp., $\mathsf{free}(\varphi) \cap \mathrm{Vr} = \emptyset$)), the formula φ is named *agent-closed* (resp., *variable-closed*). A *sentence* is a both agent- and variable-closed formula.

3.2 Semantics

Similarly as in first order logic, 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 $e' \in \mathsf{dom}(\chi) \setminus \{e\}$. A path $\pi \in \mathrm{Pth}$ is *coherent w.r.t.* an assignment $\chi \in \mathrm{Asg}$ (*χ-coherent*, for short) in case the restriction of χ on agents can be extended into a profile in respect of which π is a coherent path. Formally, π is χ-coherent iff there exists a profile $\xi \in \mathrm{Prf}$ with $\xi_{\restriction(\mathsf{dom}(\chi)\cap\mathrm{Ag})} = \chi_{\restriction \mathrm{Ag}}$ such that π is ξ-coherent. Finally, an assignment χ is *non-blocking* in a state $s \in \mathrm{St}$ in case that for each χ-coherent history ρ starting from s and not ending in a sink-state, χ in ρ agrees with one of the triggering decisions of $\mathsf{lst}(\rho)$. Formally, an assignment χ is *non-blocking* in s iff for all χ-coherent histories $\rho \in \mathrm{Hst}$ such that $\mathsf{fst}(\rho) = s$ and $\mathsf{dc}(\mathsf{lst}(\rho)) \neq \emptyset$, there exists a decision $\delta \in \mathsf{dc}(\mathsf{lst}(\rho))$ such that $\delta(a) = \chi(a)(\rho)$, for all agents $a \in \mathsf{dom}(\chi) \cap \mathrm{Ag}$. We are ready to define the semantics of our logics.

Definition 3.2. *Semantics. - Let $\mathcal{E}$ be an extension. Then, for all* SL *formulas φ, assignments $\chi \in \mathrm{Asg}$ with $\mathsf{free}(\varphi) \subseteq \mathsf{dom}(\chi)$, and states $s \in \mathrm{St}$, the modeling relation $\mathcal{E}, \chi, s \models \varphi$ is inductively defined as follows.*

1) $\mathcal{E}, \chi, s \models p$ *iff* $\mathsf{play}(\chi_{\restriction \mathrm{Ag}}, s) \in \mathsf{pr}(p)$.
2) *Boolean operators are interpreted as usual.*
3) $\mathcal{E}, \chi, s \models \langle\!\langle x \rangle\!\rangle \varphi$ *iff there exists a strategy $\sigma \in \mathrm{Str}$ such that $\mathcal{E}, \chi[x \mapsto \sigma], s \models \varphi$.*
4) $\mathcal{E}, \chi, s \models [\![x]\!] \varphi$ *iff for all strategies $\sigma \in \mathrm{Str}$, $\mathcal{E}, \chi[x \mapsto \sigma], s \models \varphi$.*
5) $\mathcal{E}, \chi, s \models (a, x)\varphi$ *iff* $\mathcal{E}, \chi[a \mapsto \chi(x)], s \models \varphi$.
6) $\mathcal{E}, \chi, s \models \odot$ *iff χ is non-blocking from s.*

Note that the satisfaction of a sentence φ does not depend on assignments, hence we omit them and write $\mathcal{E}, s \models \varphi$.

In what follows, $(\!\langle a, x \rangle\!)\varphi$ and $(\![a, x]\!)\varphi$ are abbreviations for $(a, x)(\odot \wedge \varphi)$ and $(a, x)(\odot \rightarrow \varphi)$, respectively. The formula $(\!\langle a, x \rangle\!)\varphi$ is used to ensure non-blocking assignments when the variable x is under the scope of an existential quantification (intuitively, "a adopts x, which needs to be non-blocking and to satisfy φ"). Similarly, $(\![a, x]\!)\varphi$ forces non-blocking assignments in case x is under the scope of a universal quantification (intuitively, "a adopts x, which needs to satisfy φ in case it is non-blocking).

In the scheduler example, the ability to avoid starvations can be represented by the formula $\varphi = \langle\!\langle x \rangle\!\rangle \langle\!\langle y_1 \rangle\!\rangle \langle\!\langle y_2 \rangle\!\rangle [\![z]\!] \phi$, where $\phi = (\!\langle \mathtt{A}, x \rangle\!)(\!\langle \mathtt{P}_1, y_1 \rangle\!)(\![\mathtt{P}_2, z]\!) p_1 \wedge (\!\langle \mathtt{A}, x \rangle\!)(\!\langle \mathtt{P}_2, y_2 \rangle\!)(\![\mathtt{P}_1, z]\!) p_2$ with p_1 and p_1 being the same predicates as before. It is easy to see that for all $s \in \{\mathtt{I}, \mathtt{1}, \mathtt{2}, \mathtt{A}\}$, it holds that $\mathcal{E}_S, s \models \varphi$. In particular, the arbiter strategy consists in alternating the access to the resource between the two processes, while the processes have to request the resource at least twice. Note that φ requires a unique strategy for the arbiter in order to coordinate with both the processes independently. Therefore, in terms of ATL coalition modalities, φ is weaker than $\langle\!\langle \{\mathtt{A}\} \rangle\!\rangle (p_1 \wedge p_2)$, but stronger than $\langle\!\langle \{\mathtt{A}, \mathtt{P}_1\} \rangle\!\rangle p_1 \wedge \langle\!\langle \{\mathtt{A}, \mathtt{P}_2\} \rangle\!\rangle p_2$. Actually, φ cannot be expressed in ATL*.

Once the sets of agents and predicates are fixed, SL induces a schema $\mathcal{S}_{\mathrm{SL}}$, where the set of targets is SL itself, *i.e.*, its sentences, and the schema relation is given by the definition of the semantics. A *strategy game* is a game *w.r.t.* $\mathcal{S}_{\mathrm{SL}}$. The set of all strategy games is denoted for short by $\mathrm{SG} \triangleq \mathrm{Gm}(\mathcal{S}_{\mathrm{SL}})$. We now introduce some restrictions of SG. First, we consider the case in which the numbers $|\mathrm{Ag}|$ of agents and $|\mathrm{Vr}|$ of variables that are used to write a formula are fixed to the a priori values $n, m \in [1, \omega[$. We name these two fragments SG[nAG] and SG[mVAR], respectively. Also, by SG[TB] we refer to games whose arena is turn-based. Finally, SG[Σ_k] (resp., SG[Π_k]) denotes those games where the target has a quantification prefix of k alternation starting with an existential (resp., universal) quantifier.

3.3 Binding Fragments

We now consider three fragments of SL that allow to formalize interesting game properties not expressible in ATL*. For example, they can express that an agent can join two or more different coalitions without producing mutual conflicts or that a winning condition is weaker than another one.

In what follows a *quantification prefix* over a set $\mathrm{V} \subseteq \mathrm{Vr}$ of variables 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 $\wp$. A *binding prefix* over V is a word $\flat \in \{(a, x), (\!\langle a, x \rangle\!), (\![a, x]\!) : a \in \mathrm{Ag} \wedge x \in \mathrm{V}\}^{|\mathrm{Ag}|}$ such that each agent in Ag occurs exactly once in $\flat$. By Bn we indicate the set of all binding prefixes.

Definition 3.3. *Binding Fragments.* - SL[BG] *formulas are defined by the following context-free grammar:*

$$\begin{aligned} \varphi &:= \wp\phi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi, \\ \phi &:= \flat\psi \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi, \\ \psi &:= p \mid \neg\psi \mid \psi \wedge \psi \mid \psi \vee \psi. \end{aligned}$$

where $\wp \in \mathrm{Qn}(\mathsf{free}(\psi))$, $\flat \in \mathrm{Bn}$, and $p \in \mathrm{Pr}$. Moreover, the fragments obtained by replacing the second row with $\phi := \flat\psi$, $\phi := \flat\psi \mid \phi \wedge \phi$, or $\phi := \flat\psi \mid \phi \vee \phi$, are called SL[1G], SL[CG], SL[DG]*, respectively.*

In what follows, we also consider some constraints on the number of agents/variables or the type of arenas and predicates. More specifically, TB indicates that we restrict to turn-based arenas, and kVAR (resp. kAG) bounds the maximal number of variables (resp. agents) in a formula to k. Complexity classes restrict the satisfaction problem of the predicates occurring in a formula. Finally, a slash / is used to consider the union of two fragments. For example, SG[2CG, 3AG, Σ_1] is the fragment of SL[CG] with two goals, at most 3 agents, and NPTIME-HARD predicates, whereas SG[1G, TB, 2AG/VAR] is the fragment of SG[1G] restricted to turn-based arenas and with at most two agents or variables.

Two fragments of sentences L_1 and L_2 can be compared in terms of their expressiveness. Formally, two sentences φ_1

and φ_2 are equivalent just in the case that, for all extensions $\mathcal{E}$ and $s \in \mathrm{St}$, it holds that $\mathcal{E}, s \models \varphi_1$ iff $\mathcal{E}, s \models \varphi_2$. Then, we say that L_2 is *at least as expressive as* L_1, in symbols $L_1 \preceq L_2$, if every sentence $\varphi_1 \in L_1$ is equivalent to some sentence $\varphi_2 \in L_2$. If $L_1 \preceq L_2$, but $L_2 \not\preceq L_1$, then L_2 is *more expressive than* L_1, in symbols $L_1 \prec L_2$.

Clearly, we have that SL[BG] is at least as expressive as both SL[CG] and SL[DG], which in turn are at least as expressive as SL[1G]. However, since SL[CG] and SL[DG] are not closed under negation, while this is the case for SL[BG] and SL[1G], we have that the two inclusions are necessarily strict.

THEOREM 3.1. *Expressiveness Hierarchy. - The following expressiveness relations hold:* SL[1G] $\prec$SL[CG]*,* SL[1G] $\prec$SL[DG]*,* SL[CG] $\prec$SL[BG] *and* SL[DG] $\prec$SL[BG]*.*

Since our framework trivially subsumes classic reachability games [4], we immediately obtain an hardness result *w.r.t.* the size of the arena.

THEOREM 3.2. *Arena Lower Bound. - The fulfilling problem for* SG[1G, TB, 2AG, 2VAR] *is* PTIME-HARD *in the size of the arena.*

We can also obtain hardness results *w.r.t.* the size of schemas. In particular, due to the first-order power of SL, we can reduce the truth problem of a QBF sentence to the fulfilling of a game on a fixed arena. Notably, the predicates used in the reduction can be evaluated in constant-time.

THEOREM 3.3. *Target Lower Bound. - The fulfilling problems for* SG[CG, 3AG, Σ_k] *and* SG[DG, 3AG, Π_k] *are* $\Sigma^{\mathrm{P}}_{k+1}$-HARD *and* Π^{P}_{k+1}-HARD*, respectively.*

PROOF (SKETCH). We only sketch the proof for SG[CG], since SG[DG] is just the dual case.

Let $\alpha = \wp \bigwedge_{i=1}^{n}(l_i^1 \vee l_i^2 \vee l_i^3)$ be a Σ_k QBF sentence, where, *w.l.o.g.*, the matrix is in 3CNF. Consider an arena having as agents $\mathrm{Ag} = \{1, 2, 3\}$, actions $\mathrm{Ac} = \{0, 1\}$, and states $\mathrm{St} = \{\mathtt{I}\} \cup \mathrm{D}$, where $\mathrm{D} = \{\delta \in \mathrm{Dc} : \mathsf{dom}(\delta) = \mathrm{Ag}\}$ contains the possible eight total decisions, which are seen as sink-states. In addition, if $\delta \in \mathrm{D}$ then $\mathsf{tr}(\delta)(\mathtt{I}) = \{\delta\}$ else $\mathsf{tr}(\delta)(\mathtt{I}) = \emptyset$.

We construct an SG[CG] game $\eth = \langle \mathcal{E}, \mathtt{I}, \varphi \rangle$ that is fulfilled iff α is true. The extension is built on the n predicates $p_1, \ldots, p_n$, whose valuations are set as follows: $\mathsf{pr}(p_i) \triangleq \{\mathtt{I} \cdot \delta : \delta \models l_i^1 \vee l_i^2 \vee l_i^3\}$. With $\delta \models l_i^1 \vee l_i^2 \vee l_i^3$ we mean that the formula $l_i^1 \vee l_i^2 \vee l_i^3$ is evaluated to true when assigning the values $\delta(j)$ to the variables x_i^j of the literal l_i^j. It only remains to define the target[1]: $\varphi = \wp \bigwedge_{i=1}^{n} \flat_i p_i$, where $\flat_i = (1, x_i^1)(2, x_i^2)(3, x_i^3)$. At this point, it is easy to see that $\mathcal{E}, \mathtt{I} \models \varphi$ iff α is true. Hence, the thesis is derived from the fact that the truth problem for Σ_k QBF sentences is $\Sigma^{\mathrm{P}}_{k+1}$-HARD. $\Box$

4. GAME-TYPE CONVERSIONS

In this section, we describe two game-type conversions that can be used to directly solve strategy games, provided that the predicates of the extensions are decidable. Furthermore, they substantially improve and simplify the already known decision procedures for classic SL[1G], SL[CG], and SL[DG]. In particular, *w.r.t.* [10,12], we can avoid the computational overhead resulting from the use of an automata-theoretic approach. This fact makes even clearer that, in our framework, the concept of game is considered, at the same time, as the object of study and the technical tool to analyze it.

[1]Here we are making an abuse of notation since the quantification prefix should be modified to be an SL one.

4.1 From SG[1G] to SG[1G, TB, 2AG/VAR]

First we recall that an important property used as a foundation aspect of all known elementary decision procedures for the fragments of SL is the *behavioral semantics.* Roughly speaking, a strategy logic has this property whenever the choices of an existential quantified strategy does not depend on how the other universal strategies will behave in the future or in a counterfactual play [12]. The original proof of equivalence between the classic and behavioral semantics of SL[1G] [10] is based on a quite complex reduction from the verification problem of the behavioral modeling relation $\mathcal{E}, s \models_B \wp\flat\psi$ to the solution of a classic form of two-player turn-based games. In doing so, the size of the arena is exponential in the number of actions of the original arena and doubly exponential in the number of universal variable from which an existential one depends.

Here, we propose a new reduction from generic SG[1G] to two-agent turn-based SG[1G]. Differently from previous reductions, the quantifications over actions are not treated in a one-shot fashion, but rather are emulated by finite games between the two players, the first choosing the value of the existential variables and the second the universal ones. This results in an arena just polynomial in the number of actions and exponential in the dependences between the variables.

THEOREM 4.1. *One-Goal Reduction. - For a* SG[1G, kVAR] *Borelian* $\eth$ *of order* n*, there is a* SG[1G, TB, 2AG/ VAR] *Borelian* $\eth^\star$ *of order* $n \cdot 2^{\mathrm{O}(k)}$ *such that* $\eth$ *is fulfilled iff* $\eth^\star$ *is.*

PROOF (SKETCH). Let $\eth = \langle \mathcal{E}, s, \varphi \rangle$, with target $\varphi = \wp\flat\psi$, extension $\mathcal{E} = \langle \mathcal{A}, \mathrm{Pr}, \mathsf{pr} \rangle$, and arena $\mathcal{A} = \langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \mathsf{tr} \rangle$, we show how to construct $\eth^\star = \langle \mathcal{E}^\star, \varphi^\star, t^\star \rangle$, with extension $\mathcal{E}^\star = \langle \mathcal{A}^\star, \mathrm{Pr}^\star, \mathsf{pr}^\star \rangle$ and arena $\mathcal{A}^\star = \langle \mathrm{Ag}^\star, \mathrm{Ac}^\star, \mathrm{St}^\star, \mathsf{tr}^\star \rangle$.

We start with $\mathcal{A}^\star$. It has two agents, $\exists$ and $\forall$, the former trying to prove that $\mathcal{E}, s \models \varphi$, while the latter the opposite. To achieve their task, for each state in $\mathcal{A}$, they give an evaluation to the existential and universal variables, respectively, by choosing an appropriate action. Following this idea, we set $\mathrm{Ag}^\star \triangleq \{\exists, \forall\}$ and $\mathrm{Ac}^\star \triangleq \mathrm{Ac}$. The state space has to maintain an information about the position in $\mathcal{A}$ together with the index of the variable that has still to be evaluated and the values already associated to the previous variables. To do this, we set $\mathrm{St}^\star \triangleq \mathrm{St} \times [0, |\wp|] \times_i (\mathrm{Vr}(\wp_{<i}) \rightharpoonup \mathrm{Ac})$. Observe that, when the game is in a state $(s, |\wp|, \zeta)$, all quantifications are already resolved and it is time to evaluate the corresponding decision by composing ζ with the binding $\flat$.

Before proceeding with the definition of the transition function, it is helpful to identify which are the active agents for each possible state (s, i, ζ). When $i < |\wp|$ points to an existential variable in $\wp$, *i.e.*, $\mathsf{type}(\wp_i) \triangleq \exists$, the unique owner of the state is $\exists$. Similarly, if $\mathsf{type}(\wp_i) \triangleq \forall$, the active agent is $\forall$. In the case $i = |\wp|$, instead, there are no more choices to do, so, the related state is deterministic, *i.e.*, it has no active agents. Formally, for all $(s, i, \zeta) \in \mathrm{St}^\star$, we have that if $i < |\wp|$ then $\mathsf{ag}((s, i, \zeta)) \triangleq \{\mathsf{type}(\wp_i)\}$ else $\mathsf{ag}((s, i, \zeta)) \triangleq \emptyset$.

The transition function is defined as follows. For each state (s, i, ζ) with $i < |\wp|$ and decision $\mathsf{type}(\wp_i) \mapsto c$, we simply need to increase the counter i and associate the variable $\mathsf{vr}(\wp_i)$ of $\wp_i$ with action c. Formally, we set $\mathsf{tr}^\star(\mathsf{type}(\wp_i) \mapsto c)((s, i, \zeta)) \triangleq (s, i + 1, \zeta[\mathsf{vr}(\wp_i) \mapsto c])$. For a state $(s, |\pi|, \zeta)$, instead, we just define a transition to the state $(s', 0, \varnothing)$, where s' is the successor of s in the arena $\mathcal{A}$ following the decision $\zeta \circ \flat$, whenever active. Formally, we have $\mathsf{tr}^\star(\varnothing) \triangleq \{(s, |\wp|, \zeta) \mapsto (\mathsf{tr}(\zeta \circ \flat)(s), 0, \varnothing) : \zeta \circ \flat \in \mathsf{dc}(s)\}$.

Now, we define the extension $\mathcal{E}^\star$. The predicates and their path valuations are simply inherited from the original extension $\mathcal{E}$, *i.e.*, $\mathrm{Pr}^\star \triangleq \mathrm{Pr}$ and $\pi^\star \in \mathsf{pr}^\star(p)$ iff $\pi \in \mathsf{pr}(p)$, where $\pi \in \mathrm{Pth}$ is the unique path in $\mathcal{A}$ such that $\pi^\star_i = (\pi_i, 0, \varnothing)$. Intuitively, a path $\pi^\star$ satisfies a predicate $p^\star$ iff its projection π on the states of $\mathcal{A}$ does the same.

Finally, the initial state is $s^\star \triangleq (s, 0, \varnothing)$ and the target is $\varphi^\star \triangleq \langle\!\langle x \rangle\!\rangle [\![y]\!] (\!\langle \exists, x \rangle\!) (\![\forall, y]\!) \psi$.

At this point, by means of Martin's determinacy theorem for Borelian games [9], it is not hard to show that $\Game$ is equivalent to $\Game^\star$, *i.e.*, $\Game$ is fulfilled iff $\Game^\star$ is. $\square$

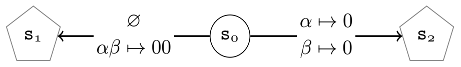


Figure 2: A full arena $\mathcal{A}$.

As an example, consider the SL[1G] game $\Game \triangleq \langle \mathcal{E}, s, \varphi \rangle$, whose underlying arena $\mathcal{A}$ is the one depicted in Figure 2, having in s_0 all decisions active and s_1, s_2 as sink-states. The extension $\mathcal{E}$ contains a unique predicate p having the path valuation $\mathsf{pr}(p) = \{s_0 \cdot s_1\}$. Moreover, the target φ is $\wp\flat p$, where $\wp = [\![x]\!] \langle\!\langle y \rangle\!\rangle$ and $\flat = (\alpha, x)(\beta, y)$. It is easy to see that $\Game$ is fulfilled, by means of the copy-cat strategy for β over α.

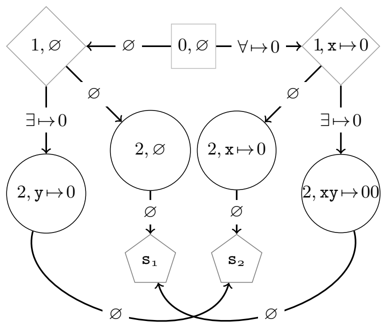


Figure 3: SL[1G] reduction of $\mathcal{A}$.

By applying the above construction, we obtain the game $\Game^\star$, whose arena is reported in Figure 3. The box state belongs to $\forall$, while the diamond ones to $\exists$. Circle and pentagonal states do not belong to any agent and, in particular, the latter are sink-state. Note that, for the sake of space, we omit the state component in all but the pentagonal nodes. Similarly, the two sink-states do not report the $0, \varnothing$ part. The path valuation of the predicate p is $\mathsf{pr}^\star(p) = \{(s_0, 0, \varnothing) \cdot (s_0, 1, \zeta_1) \cdot (s_0, 2, \zeta_2) \cdot (s_1, 0, \varnothing) : \zeta_1 = \zeta_2 = \varnothing \vee \zeta_1 = x \mapsto 0 \wedge \zeta_2 = xy \mapsto 00\}$. Finally, the initial state is $s^\star = (s_0, 0, \varnothing)$ and the target is $\varphi^\star \triangleq \langle\!\langle y \rangle\!\rangle [\![x]\!] (\!\langle \exists, y \rangle\!) (\![\forall, x]\!) p$. It can be seen that $\Game^\star$ is fulfilled, as well. Indeed, $\exists$ can choose $\varnothing$ on $(s_0, 1, \varnothing)$ and $\exists \mapsto 0$ on $(s_0, 1, \mathrm{x} \mapsto 0)$, forcing the play to reach $(s_1, 0, \varnothing)$.

4.2 From SG[CG/DG, kVAR] to SG[1G, $(k+1)$AG/VAR]

Recently, it has been proved that the two SL[1G] extensions SL[CG] and SL[DG] enjoy the behavioral semantics as well [12]. Actually, in the hierarchy obtained by restricting the combination of goals, they represent the maximal fragments of SL satisfying such a property. Indeed, there are SL[BG] satisfiable sentences that are not behaviorally satisfiable [10].

As in the case of SL[1G], the proof for these fragments was a direct reduction to a two-player turn-based game. However, its structure was much more complex as it further required to keep memory in the states of the information about the current position of each binding. This fact results in a state space whose cardinality was $\mathrm{O}(|\mathrm{St}|^h)$, where h is the number of bindings used in the related sentence.

Here, we propose a completely different approach. We reduce the fulfilling of a conjunctive/disjunctive game to the same problem of a concurrent one-goal game. In particular, the construction simulates the conjunction (resp., disjunction) of bindings by means of a dedicated new agent, whose relative strategy is universally (resp., existentially) quantified. By doing so, we can avoid to keep memory of all positions at the same time, obtaining a state space whose size is $\mathrm{O}(|\mathrm{St}| \cdot 2^h)$.

Theorem 4.2. *Conjunctive/Disjunctive-Goal Reduction. - For each* SG[hCG/DG, kVAR] $\Game$ *of order* n*, there is an* SG[1G, $(k+1)$ AG/VAR] $\Game^\star$ *of order* $\mathrm{O}(n \cdot 2^h)$ *such that* $\Game$ *is fulfilled iff* $\Game^\star$ *is.*

Proof (Sketch). Given an SG[CG] (resp., SG[DG]) $\Game = \langle \mathcal{E}, s, \varphi \rangle$, with target $\varphi = \wp \bigwedge_{\flat \in \mathrm{B}} \flat\psi_\flat$ (resp., $\varphi = \wp \bigvee_{\flat \in \mathrm{B}} \flat\psi_\flat$), extension $\mathcal{E} = \langle \mathcal{A}, \mathrm{Pr}, \mathsf{pr} \rangle$, and arena $\mathcal{A} = \langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \mathsf{tr} \rangle$, we show how to construct $\Game^\star = \langle \mathcal{E}^\star, \varphi^\star, t^\star \rangle$, with extension $\mathcal{E}^\star = \langle \mathcal{A}^\star, \mathrm{Pr}^\star, \mathsf{pr}^\star \rangle$ and arena $\mathcal{A}^\star = \langle \mathrm{Ag}^\star, \mathrm{Ac}^\star, \mathrm{St}^\star, \mathsf{tr}^\star \rangle$.

We start with $\mathcal{A}^\star$. Its agents are the free variables of $\phi = \circledast_{\flat \in \mathrm{B}} \flat\psi_\flat$, with $\circledast \in \{\wedge, \vee\}$, *i.e.*, those quantified in $\wp$, extended with the fresh agent $\circledast$, whose role is to simulate the corresponding Boolean combination of goals: $\mathrm{Ag}^\star \triangleq \mathsf{free}(\phi) \cup \{\circledast\}$. The actions are the ones in $\mathcal{A}$ augmented with a fresh one for each binding prefix in B. The latter lets the agent $\circledast$ to choose which goal to verify: $\mathrm{Ac}^\star \triangleq \mathrm{Ac} \cup \mathrm{B}$. The state space is just the Cartesian product of St with the power-set of the binding prefixes set B, as we have to keep memory on which goals are still active in a given state: $\mathrm{St}^\star \triangleq \mathrm{St} \times (2^{\mathrm{B}} \setminus \{\emptyset\})$.

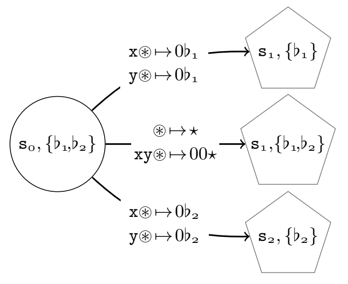
Before defining the transition function, it is useful to determine which decisions are active in a given state. To do this, note that we need agent $\circledast$ to be active in every state $(s, \mathrm{X}) \in \mathrm{St}^\star$ and let him to choose between the binding prefixes in X that have still to be verified from s onward. Moreover, the remaining part of an active decision has to ensure that, once it is composed with the selected binding prefix, it returns an active decision of $\mathcal{A}$ in s. Formally, we set $\mathsf{dc}((s, \mathrm{X})) \triangleq \{\delta^\star \in \mathrm{Dc}^\star : \circledast \in \mathsf{dom}(\delta^\star) \wedge \delta^\star(\circledast) \in \mathrm{X} \wedge \delta^\star \circ \delta^\star(\circledast) \in \mathsf{dc}(s)\}$.

We now set the transition function $\mathsf{tr}^\star$. Given an active decision $\delta^\star \in \mathsf{dc}((s, \mathrm{X}))$ on a state $(s, \mathrm{X}) \in \mathrm{St}^\star$, the successor state (s', X') is set as follows. The component s' is the successor of s following the decision $\delta^\star \circ \delta^\star(\circledast)$ obtained by assigning the value of the variables to the agents of $\mathcal{A}$ by means of the binding prefix $\delta^\star(\circledast)$ chosen by $\circledast$. The component X' is the subset of X containing all binding prefixes that, once selected, lead from s to s'. Formally, we have that $\mathsf{tr}^\star(\delta^\star)((s, \mathrm{X})) \triangleq (\mathsf{tr}(\delta^\star \circ \delta^\star(\circledast))(s), \{b \in \mathrm{X} : \mathsf{tr}(\delta^\star \circ \delta^\star(\circledast))(s) = \mathsf{tr}(\delta^\star \circ b)(s)\})$.

Now, we define the extension $\mathcal{E}^\star$. The paths $\pi^\star \in \mathrm{Pth}^\star$ in $\mathcal{A}^\star$ for a goal $\flat\psi_\flat$ are those having $\flat$ as element of the second component of all its states, *i.e.*, $\flat \in \mathrm{X}_i$, for all $i \in [0, |\pi^\star|[$, where $(\pi^\star)_i = (s_i, \mathrm{X}_i)$. We denote the set of all these paths by $\mathrm{P}_\flat \subseteq \mathrm{Pth}^\star$. We consider a unique predicate $p^\star$ representing the paths that satisfy the formulas $\psi_\flat$, for each binding prefix $\flat$ of interest. This means that, we first set $\mathrm{Pr}^\star \triangleq \{p^\star\}$. Then, for all $\pi^\star \in \mathrm{Pth}^\star$, we define the predicate function $\mathsf{pr}^\star$ as follows, where $\pi \in \mathrm{Pth}$ is obtained from $\pi^\star$ by projecting out the second component of all its states. If $\circledast = \wedge$, then $\pi^\star \in \mathsf{pr}^\star(p^\star)$ iff, for each $\flat \in \mathrm{B}$ with $\pi^\star \in \mathrm{P}_\flat$, it holds that $\pi \in \mathsf{pr}(\psi_\flat)$. If $\circledast = \vee$, instead, $\pi^\star \in \mathsf{pr}^\star(p^\star)$ iff there exists $\flat \in \mathrm{B}$ with $\pi^\star \in \mathrm{P}_\flat$ such that $\pi \in \mathsf{pr}(\psi_\flat)$.

The initial state $s^\star$ is set as (s, B). Finally, we define the target $\varphi^\star$. The quantification prefix $\wp^\star$ is the juxtaposition of the prefix $\wp$ with either a universal or an existential quantifier over the fresh variable $\circledast$, depending on whether $\circledast$ is a conjunction or a disjunction. As binding prefix $\flat^\star$, we use the one that associates with all agents the variables having the same names. In particular, the type of each binding depends on the type of quantification. Hence, the target is $\varphi^\star = \wp^\star \flat^\star p^\star$. $\square$

As an example of the above reduction, consider the SL[CG] game $\eth = \langle \mathcal{E}, s_0, \varphi \rangle$, whose underlying arena $\mathcal{A}$ is again the one depicted in Figure 2. In the extension $\mathcal{E} = \langle \mathcal{A}, \mathrm{Pr}, \mathsf{pr} \rangle$, there are two predicates p_1 and p_2 such that $\mathsf{pr}(p_i) = \{s_0 \cdot s_i\}$, for all $i \in \{1, 2\}$. The target φ is the sentence $[\![x]\!] \langle\!\langle y \rangle\!\rangle \bigwedge_{i=1}^{2} \flat_i p_i$, where $\flat_1 = (\alpha, y)(\beta, y)$ and $\flat_2 = (\alpha, x)(\beta, y)$. It is easy to see that $\eth$ is fulfilled.



$\mathtt{x}\circledast \mapsto 0\flat_1$
$\mathtt{y}\circledast \mapsto 0\flat_1$
$\mathtt{s}_1, \{\flat_1\}$
$\circledast \mapsto \star$
$\mathtt{xy}\circledast \mapsto 00\star$
$\mathtt{s}_0, \{\flat_1, \flat_2\}$
$\mathtt{s}_1, \{\flat_1, \flat_2\}$
$\mathtt{x}\circledast \mapsto 0\flat_2$
$\mathtt{y}\circledast \mapsto 0\flat_2$
$\mathtt{s}_2, \{\flat_2\}$

Figure 4: SL[CG/DG] **reduction of** $\mathcal{A}$.

Now, by applying the construction described above, we obtain an SL[1G] game $\eth^\star$ whose arena, represented in Figure 4, has $(s_0, \{\flat_1, \flat_2\})$ as the unique state with active agents. The unique predicate $p^\star$ is associated with the following path valuation: $\mathsf{pr}^\star(p^\star) = \{(s_0, \{\flat_1, \flat_2\}) \cdot (s_1, \{\flat_1\}), (s_0, \{\flat_1, \flat_2\}) \cdot (s_2, \{\flat_2\})\}$. The initial state of $\eth^\star$ is $(s_0, \{\flat_1, \flat_2\})$ and the target $\varphi^\star$ is the sentence $\wp^\star \flat^\star p^\star$, where $\wp^\star = [\![x]\!] \langle\!\langle y \rangle\!\rangle [\![\circledast]\!]$ and $\flat^\star = (\![x, x]\!) (\!\langle y, y \rangle\!) (\![\circledast, \circledast]\!)$. It is not hard to see that also $\eth^\star$ is fulfilled, as well.

5. DISCUSSION

Several logics for strategic reasoning such as ATL* and SL borrows LTL operators to specify temporal properties over the evolutions of a game. This seemed a natural choice, since LTL allows to specify a widely range of conditions such as fairness, reachability, safety and so on. Nevertheless, some contexts may require more expressive languages than LTL or, conversely, if someone is interested in a specific game, say chess, it could be profitable to implement an ad-hoc procedure to check checkmates or stalemates. Summarily, LTL is not the all-inclusive solution to the verification of temporal properties.

For this reason, we revisit the classical logic of strategic reasonings and propose a new game framework which focuses on the "pure" strategic components, leaving the problem to check temporal properties to an external oracle. This, on the one hand, enables to isolate the contribution of the solely strategic modalities to the model-checking problem. In particular, we show new lower bounds which make very minimal assumptions on the complexity of the temporal part. As a side effect, the proposed framework has allowed to readdress and substantially improve previous complexity results for the SL fragments SL[1G], SL[CG], and SL[DG]. In particular for SL[1G], the exponential dependence on the number of actions occurring in formula decreases to a polynomial one and the double exponential dependence on the number of dependences between variables to a single exponential. For SL[CG] and SL[DG], instead, we have a polynomial but significant improvement in the order of the game we need to solve, since we pass from a $\mathrm{O}(|\mathrm{St}|^h)$ to $\mathrm{O}(|\mathrm{St}| \cdot 2^h)$, where h is the number of agent bindings.

Acknowledgments

We thank M. Benerecetti for useful discussions on the reduction from conjunctive/disjunctive-goal to one-goal games.

This paper is partially supported by the FP7 EU project 600958-SHERPA and the Italian projects IndAM "Logiche di Gioco Estese", PON OR.C.HE.S.T.R.A., and POR Embedded System Cup B25B09090100007.

6. REFERENCES

[1] T. Ågotnes, V. Goranko, and W. Jamroga. Alternating-Time Temporal Logics with Irrevocable Strategies. In *TARK'07*, pages 15–24, 2007.

[2] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-Time Temporal Logic. *JACM*, 49(5):672–713, 2002.

[3] K. Chatterjee, T.A. Henzinger, and N. Piterman. Strategy Logic. *IC*, 208(6):677–693, 2010.

[4] N. Immerman. Number of Quantifiers is Better Than Number of Tape Cells. *JCSS*, 22(3):384–406, 1981.

[5] W. Jamroga and W. van der Hoek. Agents that Know How to Play. *FI*, 63(2-3):185–219, 2004.

[6] 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.

[7] 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.

[8] E. Lorini. A Dynamic Logic of Agency II: Deterministic DLA, Coalition Logic, and Game Theory. *JLLI*, 19(3):327–351, 2010.

[9] A.D. Martin. Borel Determinacy. *AM*, 102(2):363–371, 1975.

[10] F. Mogavero, A. Murano, G. Perelli, and M.Y. Vardi. Reasoning About Strategies: On the Model-Checking Problem. Technical report, arXiv, 2011.

[11] 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.

[12] F. Mogavero, A. Murano, and L. Sauro. On the Boundary of Behavioral Strategies. In *LICS'13*, pages 263–272. IEEE Computer Society, 2013.

[13] F. Mogavero, A. Murano, and M.Y. Vardi. Reasoning About Strategies. In *FSTTCS'10*, LIPIcs 8, pages 133–144. Leibniz-Zentrum fuer Informatik, 2010.

[14] D. Perrin and J. Pin. *Infinite Words*. Pure and Applied Mathematics. Elsevier, 2004.

[15] A. Pnueli. The Temporal Logic of Programs. In *FOCS'77*, pages 46–57. IEEE Computer Society, 1977.

[16] 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.

[17] J. van Eijck. PDL as a Multi-Agent Strategy Logic. In *TARK'13*, pages 206–215, 2013.

[18] M.Y. Vardi. A Temporal Fixpoint Calculus. In *POPL'88*, pages 250–259. Association for Computing Machinery, 1988.

[19] D. Walther, W. van der Hoek, and M. Wooldridge. Alternating-Time Temporal Logic with Explicit Strategies. In *TARK'07*, pages 269–278, 2007.

[20] P. Wolper. Temporal Logic Can Be More Expressive. *IC*, 56(1-2):72–99, 1983.