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* $\eth \triangleq \langle \mathcal{E}, s, t \rangle \in \mathrm{Gm}(\mathcal{S})$*, 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 $\eth \triangleq \langle \mathcal{E}, s, t \rangle$ 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. $\square$

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 $\varphi^{\star} \triangleq \langle\!\langle x \rangle\!\rangle [\![y]\!] (\![\exists, x)\!) (\![\forall, y)\!) \psi$.

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. □

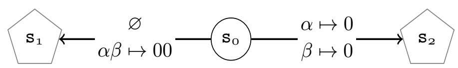

Figure 2: A full arena $\mathcal{A}$.

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. 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 $\varphi^{\star} \triangleq \langle\!\langle y \rangle\!\rangle [\![x]\!] (\![\exists, y)\!) (\![\forall, x)\!) p$. 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 $(s_0, 1, \mathtt{x} \mapsto 0)$, forcing the play to reach $(s_1, 0, \varnothing)$.

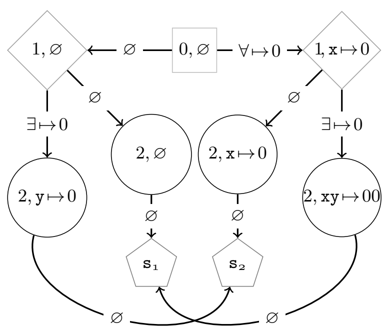

Figure 3: SL[1G] reduction of $\mathcal{A}$.

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 $\varphi^\star = \wp^\star \flat^\star p^\star$. □

As an example of the above reduction, consider the SL[CG] game $\Game = \langle \mathcal{E}, s_0, \varphi \rangle$, 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 $\Game$ is fulfilled.

Now, by applying the construction described above, we obtain an SL[1G] game $\Game^\star$ 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 $\Game^\star$ 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 $\Game^\star$ is fulfilled, as well.

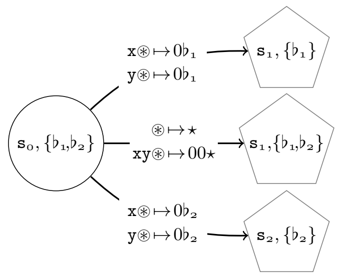

Figure 4: SL[CG/DG] **reduction of** $\mathcal{A}$**.**

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.