Bisimulations for Logics of Strategies: A Study in Expressiveness and Verification

Francesco Belardinelli, Catalin Dima, Aniello Murano

Proceedings of the Sixteenth International Conference on
Principles of Knowledge Representation and Reasoning (KR 2018)

Bisimulations for Logics of Strategies: A Study in Expressiveness and Verification

Francesco Belardinelli
Université d'Evry
France

Catalin Dima
Université Paris-Est Créteil
France

Aniello Murano
Università di Napoli Federico II
Italy

Abstract

In this paper we advance the state of the art on the subject of bisimulations for logics of strategies. Bisimulations are a key notion to study the expressive power of a modal language, as well as for applications to system verification. In this contribution we present novel notions of bisimulation for several significant fragments of Strategy Logic (SL), and prove that they preserve the interpretation of formulas in the corresponding fragments. In selected cases we are able to prove that such bisimulations enjoy the Hennessy-Milner property. Finally, we make use of bisimulations to study the expressiveness of the various fragment of SL, including the complexity of their model checking problems.

1 Introduction

Logics for strategies have acquired preeminence in recent years in the specification and verification of complex strategic abilities in multi-agent scenarios. Amongst these modal languages we mention Alternating-time Temporal Logic (Alur, Henzinger, and Kupferman 2002), Strategy Logic (Chatterjee, Henzinger, and Piterman 2010; Mogavero et al. 2014), Coalition Logic (Pauly 2002). In combination with tools and techniques for formal verification by model checking (Kacprzak et al. 2008; Lomuscio, Qu, and Raimondi 2017; Cermák, Lomuscio, and Murano 2015), these logic-based languages are rightly considered as one of the success stories in the applications of formal methods to reasoning about strategic behaviours of autonomous agents in game structures and multi-agent systems.

As it is the case for any modal language, bisimulation relations are a key notion to study the expressive power of logics for strategies, beginning with van Benthem's characterisation of propositional modal logic as the bisimulation-invariant fragment of first-order logic (van Benthem 1976). Further, bisimulations are crucial for system verification. Indeed, whenever we are confronted with a model checking problem $M \models \phi$ that is not amenable to practical verification, we might consider to replace model M with a *bisimilar*, possibly smaller model M'. Then, we can solve the new model checking problem $M' \models \phi$, and finally transfer the result thus obtained to the original M in virtue of a preservation result. Over the years, several abstractions and refinement techniques have appeared to build compact bisimulations efficiently (Clarke et al. 2000; Belardinelli, Lomuscio, and Michaliszyn 2016).

In this contribution we advance the state of the art on the subject of bisimulations for several significant fragments of Strategy Logic, and show the preservation of formulas in the relevant languages. In selected cases we are able to prove that such bisimulations enjoy the Hennessy-Milner property. Then, we make use of bisimulations to study the expressiveness of the various fragments of SL, including the complexity of their model checking problems. Finally, we illustrate the usefulness of our model-theoretic results by showing that the elimination of dominated strategies, a game-theoretic notion, generates bisimilar game structures.

Related work. The literature on bisimulations for modal logics is extensive, a survey of model equivalences for various temporal logics appears in (Goltz, Kuiper, and Penczek 1992). The landscape for logics of strategic abilities is comparatively more sparse. Alternating bisimulations for ATL^* were introduced in (Alur et al. 1998), including a proof of the Hennessy-Milner property. As far as we know, no attempt has been made so far to extend this notion to more expressive languages such as fragments of Strategy Logic. Indeed, (Mogavero 2013) shows that Strategy Logic itself is not preserved by standard bisimulation relations, but rather by the fairly stronger local isomorphisms. Also (Gutierrez et al. 2017) studies the preservation of Nash Equilibria by labeled bisimulation, but does not take into account other variants of bisimulation (like alternating bisimulation). Note that our result from Section 4.2 does not compare with the negative result from (Gutierrez et al. 2017), see the discussion after Lemma 29. In a different direction, there have been several attempts to extend alternating bisimulations for ATL^* to contexts of imperfect information (Ågotnes, Goranko, and Jamroga 2007; Dastani and Jamroga 2010; Belardinelli et al. 2017). Differently from these contributions, here we consider formal languages strictly stronger than ATL^*, interpreted on systems with perfect information. To the best of our knowledge, ours are the first results on bisimulations in such a setting.

Scheme of the paper. In Sec. 2 we introduce Strategy Logic (SL) and several of its fragments. Then, we provide them with a semantics in terms of Concurrent Game Structures (CGS). We compare these fragments w.r.t. the state of the art. In particular, we prove that fragment SSL^- to be introduced is as expressive as ATL^* with strategy contexts (Laroussinie and Markey 2015). In Sec. 3 and 4 we define novel, truth-preserving bisimulations for all fragments of SL considered. The key difference here is that, while the bisimulations in Sec. 3 enjoy the Hennessy-Milner property, those in Sec. 4 dont't. Then, in Sec. 5 we analyse the expressive power of our fragments, as well as the complexity of their model checking problems. Further, we show that the elimination of dominated strategies generate bisimilar game structures. We conclude in Sec. 6 by pointing to future work.

For reasons of space, the lengthier proofs are available in a separate document submitted with the paper.

2 Logics for Strategies

In this section we introduce Strategy Logic (SL) and its fragments. These languages are then interpreted on concurrent game structures (CGS), as customary. We start with some notation. For a finite or infinite non-empty sequence $u \in X^\omega \cup X^+$ of elements in some set X, we write u_i for the $(i+1)$-th element of u, i.e., $u = u_0 u_1 \ldots$. For $i \geq 0$, $u_{\leq i}$ is the prefix of u of length $i+1$, i.e., $u_{\leq i} = u_0 u_1 \ldots u_i$. The empty sequence is denoted as ϵ. The length of a finite sequence $u \in X^*$ is denoted as $|u|$, and its last element $u_{|u|-1}$ as $last(u)$.

2.1 Strategy Logic and its Fragments

For the rest of the paper we fix an infinite set *AP* of *atomic propositions (atoms)*, a finite set *Ag* of *agents*, and an infinite set *Var* of variables $x_0, x_1, \ldots$ for strategies.

Definition 1 (SL (Mogavero et al. 2014))**.** *The* formulas in Strategy Logic defined over *AP*, *Ag*, and *Var are built as follows:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \langle\langle x \rangle\rangle\varphi \mid (x, a)\varphi$$

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

The *temporal operators* from LTL are *next* X and *until* U. The *strategy quantifier* $\langle\langle x \rangle\rangle$ is read as "for some strategy x, ...", and the *binding operator* (x, a) intuitively means that "by using strategy x, agent a can enforce ...". We use the standard abbreviations, e.g., $[\![x]\!]\varphi$ for $\neg\langle\langle x \rangle\rangle\neg\varphi$. Further, for any quantifier $Qx \in \{\langle\langle x \rangle\rangle, [\![x]\!]\}$ and $A = \{a_1, \ldots, a_n\} \subseteq Ag$, we write $Q\vec{x}(\vec{x}, A)\varphi$ as a shorthand for $Qx_1, \ldots, Qx_n(x_1, a_1), \ldots, (x_n, a_n)\varphi$.

We now introduce the set *free*(φ) of free agents and variables appearing in a formula φ as standard.

Definition 2 (Free agents and variables)**.** *The set free$(\varphi) \subseteq Ag \cup Var$ of free agents and variables in a formula φ is defined as follows:*

$$\begin{aligned}
free(p) &= \emptyset \\
free(\neg\varphi) &= free(\varphi) \\
free(\varphi \wedge \varphi') &= free(\varphi) \cup free(\varphi') \\
free(\mathsf{X}\,\varphi) &= Ag \cup free(\varphi) \\
free(\varphi\,\mathsf{U}\,\varphi') &= Ag \cup free(\varphi) \cup free(\varphi') \\
free(\langle\langle x \rangle\rangle\varphi) &= free(\varphi) \setminus \{x\} \\
free((x,a)\varphi) &= \begin{cases} free(\varphi) & \text{if } a \notin free(\varphi) \\ (free(\varphi) \setminus \{a\}) \cup \{x\} & \text{otherwise} \end{cases}
\end{aligned}$$

A *sentence* is a formula φ with *free*$(\varphi) = \emptyset$. Finally, we define $shr(x, \varphi) = \{a \in Ag \mid (x, a)\psi \text{ is a subformula of } \varphi\}$ as the set of agents using strategy x in evaluating φ.

Fragments. We now introduce several fragments of SL, whose relevance will be illustrated and discussed in the rest of the paper. A *binding prefix* over a set $A \subseteq Ag$ of agents and $V \subseteq Var$ of variables is a finite sequence $\flat \in \{(x, a) \mid a \in A \text{ and } x \in V\}^{|A|}$ of length $|\flat| = |A|$, such that every agent $a \in A$ occurs exactly once in $\flat$. On the other hand, the same variable $x \in V$ can occur several times in $\flat$, i.e., intuitively, the same strategy x can be used by several agents.

A *quantification prefix* over a set $V \subseteq Var$ of variables is a finite sequence $\wp \in \{\langle\langle x \rangle\rangle, [\![x]\!] \mid x \in V\}^{|V|}$ of length $|\wp| = |V|$ such that every variable $x \in V$ occurs exactly once in $\wp$. Finally, $Qnt(V) = \{\langle\langle x \rangle\rangle, [\![x]\!] \mid x \in V\}^{|V|}$ and $Bnd(A) = \{(x, a) \mid a \in A \text{ and } x \in Var\}^{|A|}$ denote, respectively, the sets of all quantification and binding prefixes over variables in V and agents in A.

We now introduce the well-known *One-Goal* fragment of Strategy Logic (Mogavero et al. 2012; 2012).

Definition 3 (SL[1G])**.** *The formulas in* SL[1G] *defined over AP, Ag, and Var are built as follows:*

$$\varphi \quad ::= \quad p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \wp\flat\varphi$$

where $\flat \in Bnd(Ag)$ and $\wp \in Qnt(free(\flat\varphi))$.

Intuitively, in SL[1G] each goal, i.e., each formula obtained by using LTL operators X and U, preceded by a binding prefix $\flat$ for all agents in *Ag*, and by a quantification prefix $\wp$ for all strategy variables used by the agents. Because of this feature (as well as the complexity of the model checking problem), SL[1G] is considered a natural extension of the Alternating-time Temporal Logic ATL^* (Alur, Henzinger, and Kupferman 2002) to arbitrary quantification over the agents' strategies (Mogavero et al. 2012; Cermák, Lomuscio, and Murano 2015)

Here we extend this idea by allowing arbitrary binding prefixes in our formulas. Specifically, while SL and SL[1G] have already appeared in the literature, to the best of our knowledge none of the following fragments has been considered yet.

Definition 4 (SSL)**.** *The* formulas in Strict SL defined over *AP*, *Ag*, and *Var are built as follows:*

$$\varphi \quad ::= \quad p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \wp\flat\varphi$$

where $\flat \in Bnd(A)$ for some $A \subseteq Ag$ and $\wp \in Qnt(free(\flat\varphi))$.

Strict SL "generalises" SL[1G] in the sense that a binding prefix $\flat$ might refer to a strict subset A of *Ag*. Differently from SL, every variable quantified in $\wp$ is "immediately" assigned to some agent in $\flat$, similarly to SL[1G].

Further, we introduce the fragment SSL^-, for which the last clause in Def. 4 of SSL is restricted to binding prefixes $\flat$ in which all variables are different. That is, we cannot express different agents using the same strategy.

Finally, we define fragment $\mathsf{SSL}^-[1\mathsf{G}]$ as the intersection of SSL^- and SL[1G]. That is, Def. 3 is restricted to binding prefixes $\flat$ where all variables are different.

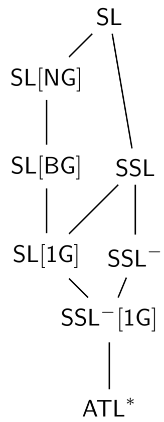

Figure 1: Inclusions between fragments.

In Fig. 1 we summarize the main language inclusions between our fragments. We include the nested-goal SL[NG] and boolean-goal SL[BG] fragments of SL for completeness, even though we will not prove any result for these. We refer to (Mogavero et al. 2014) for a formal definition. Moreover, we include ATL^* as operator $\langle\langle A\rangle\rangle$ can be expressed as $\langle\langle\vec{x}\rangle\rangle(\vec{x},A)[\![\vec{y}]\!](\vec{y},Ag\setminus A)$. Notice that SL[1G] and SSL^- are incomparable. In Sec. 5 we will prove that semantically all inclusions are strict.

2.2 Concurrent Game Structures

We now provide a semantics to the various fragments of SL introduced in Sec. 2.1 by means of concurrent game structures.

Definition 5 (CGS)**.** *Given sets Ag of agents and AP of atoms, a* concurrent game structure *is a tuple* $\mathbf{G} = \langle Ag, AP, S, s_0, \{Act_a\}_{a\in Ag}, \tau, L\rangle$ *such that*

- *S is a non-empty finite set of* states*; $s_0 \in S$ is the* initial states *of* $\mathbf{G}$.
- *For every agent $a \in Ag$, Act_a is a finite non-empty set of* actions. *A tuple $\vec{\alpha} = (\alpha_a)_{a\in Ag} \in ACT = \prod_{a\in Ag} Act_a$ is called a* joint action.
- *$\tau : S \times ACT \to S$ is the* transition function. *We often write $s \xrightarrow{\vec{\alpha}} t$ for $\tau(s,\vec{\alpha}) = t$.*
- *$L : S \to 2^{AP}$ is the* labelling function.

Given a CGS $\mathbf{G}$ as above, a *path* is a (finite or infinite) sequence $\pi \in S^* \cup S^\omega$ such that for every $j \geq 0$, $\pi_j \xrightarrow{\vec{\alpha}_j} \pi_{j+1}$ for some joint action $\vec{\alpha}_j$. We distinguish between finite paths, or *histories*, and infinite paths, or *computations*. Recall that for a path π and $j \geq 0$, $\pi_{\leq j}$ denotes the initial history of length $j+1$, and $last(h)$ is last element in history h.

Definition 6 (Strategy)**.** *A* (memoryful) strategy *for an agent $a \in Ag$, or a-strategy, is a function $\sigma : S^+ \to Act_a$.*

The set of all strategies, for all agents, is denoted as $\Sigma(\mathbf{G})$. Then, an *A-strategy* is an a-strategy for every $a \in A \subseteq Ag$, and a *joint strategy* is a function $\sigma_{Ag} : Ag \to \Sigma(\mathbf{G})$ that associates to every agent $a \in Ag$ a strategy for a. We write $\sigma_{Ag}(a) = \sigma_a$. For every $h \in S^+$, a joint strategy σ_{Ag} defines a unique computation $\lambda(h,\sigma_{Ag}) = h \xrightarrow{\sigma_{Ag}(h)} s_1 \xrightarrow{\sigma_{Ag}(h\cdot s_1)} s_2 \ldots$, that starts with h and is consistent with σ_{Ag}.

Further, an *assignment* is a function $\chi : Var \cup Ag \to \Sigma(\mathbf{G})$ such that for every agent $a \in Ag$, $\chi(a)$ is a strategy for a. For $z \in Var \cup Ag$ and $\sigma \in \Sigma(\mathbf{G})$, the *variant* χ^z_σ is the assignment that maps z to σ and coincides with χ on all other variables and agents. We often write χ^A_σ as a shorthand for $(\ldots(\chi^{a_1}_\sigma)\ldots)^{a_k}_\sigma$ for $A = \{a_1,\ldots,a_k\}$. Also, we write $\lambda(h,\chi)$ for $\lambda(h,\chi(Ag))$, that is, the unique computation starting with h and consistent with joint strategy $\chi(Ag)$.

Definition 7 (Satisfaction)**.** *We inductively define $(\mathbf{G},h,\chi) \models \varphi$ where h is a history, φ is a formula, and χ is an assignment such that for every $x \in Var$, $\chi(x)$ is a strategy for all agents in $shr(x,\varphi)$:*

$(\mathbf{G},h,\chi) \models p$ *iff* $p \in L(last(h))$
$(\mathbf{G},h,\chi) \models \neg\varphi$ *iff* $(\mathbf{G},h,\chi) \not\models \varphi$
$(\mathbf{G},h,\chi) \models \varphi_1 \wedge \varphi_2$ *iff* $(\mathbf{G},h,\chi) \models \varphi_i$ *for* $i \in \{1,2\}$
$(\mathbf{G},h,\chi) \models \mathsf{X}\,\varphi$ *iff* $(\mathbf{G},\lambda(h,\chi)_{\leq|h|+1},\chi) \models \varphi$
$(\mathbf{G},h,\chi) \models \varphi_1 \,\mathsf{U}\, \varphi_2$ *iff for some* $i \geq |h|$, $(\mathbf{G},\lambda(h,\chi)_{\leq i},\chi) \models \varphi_2$, *and for all* j, $|h| \leq j < i$ *implies* $(\mathbf{G},\lambda(h,\chi)_{\leq j},\chi) \models \varphi_1$
$(\mathbf{G},h,\chi) \models \langle\langle x\rangle\rangle\varphi$ *iff for some strategy* σ *for every agent in* $shr(x,\varphi)$, $(\mathbf{G},h,\chi^x_\sigma) \models \varphi$
$(\mathbf{G},h,\chi) \models (x,a)\varphi$ *iff* $(\mathbf{G},h,\chi^a_{\chi(x)}) \models \varphi$

We write $(\mathbf{G},h) \models \varphi$ to mean that $(\mathbf{G},h,\chi) \models \varphi$ for every assignment χ. In particular, $\mathbf{G} \models \varphi$ iff $(\mathbf{G},s_0) \models \varphi$. Then, CGS $\mathbf{G}$ and $\mathbf{G}'$ are *logically equivalent* (w.r.t. language L) if for every formula $\varphi \in \mathsf{L}$, $\mathbf{G} \models \varphi$ iff $\mathbf{G} \models \varphi$.

We state without proof that the satisfaction of formulas depends only on their free variables and agents, that is, if assignments χ and χ' coincide on $free(\varphi)$, then $(\mathbf{G},h,\chi) \models \varphi$ iff $(\mathbf{G},h,\chi') \models \varphi$. In particular, if φ is a sentence, then $(\mathbf{G},h) \models \varphi$ iff $(\mathbf{G},h,\chi) \models \varphi$ for some assignment χ.

Definition 8. *Given two CGS $\mathbf{G}$ and $\mathbf{G}'$, we say that $\mathbf{G}$ and $\mathbf{G}'$ are* logically equivalent *w.r.t. one of the above fragments of* SL *iff for any closed formula φ and any strategy profiles χ in $\mathbf{G}$ and χ' in $\mathbf{G}'$, we have that $(\mathbf{G},s_0,\chi) \models \varphi$ if and only if $(\mathbf{G}',s_0',\chi') \models \varphi$.*

Remark 9. *Consider the following grammar:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi\wedge\varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \langle\langle x\rangle\rangle(x,a_1)\ldots(x,a_k)\varphi$$

where $p \in AP$, $x \in Var$, and $a_1,\ldots,a_k \in Ag$.

We observe that every formula φ in SSL *can be rewritten as an equivalent formula φ' in the grammar above. To check this fact, notice that any* SSL *formula $\wp\flat\varphi$ with*

$$\wp = Qx_1\ldots Qx_k$$
$$\flat = (x_1,a_{1,1})\ldots(x_1,a_{1,m_1})\ldots(x_k,a_{k,1})\ldots(x_k,a_{k,m_k})$$

is equivalent to

$$Qx_1(x_1,a_{1,1})\ldots(x_1,a_{1,m_1}) \ldots Qx_k(x_k,a_{k,1})\ldots(x_k,a_{k,m_k})\varphi$$

which is indeed a formula in the grammar above.

As an immediate consequence, every formula φ in SSL^- can be rewritten as an equivalent formula φ' in the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \langle\langle x\rangle\rangle(x,a)\varphi$$

Since the two alternative grammars for SSL *and* SSL^- *are equally expressive, we will use them interchangeably in the rest of the paper, depending on which is more appropriate for the task at hand (e.g., Def. 14).*

The advantage of the present grammar is to show clearly that in SSL *and* SSL^- *strategy variables are immediately bound (i.e., assigned to agents) after being introduced through quantification.*

Example 10. *To illustrate the formal machinery introduced so far, we recall briefly the notion of game in normal form, its encoding as a CGS, and the encoding of* Nash Equilibria *(NE) in a game in normal form as a model checking problem for CGS and* SSL^-.

A game in normal form *is a tuple* $\Gamma = (Ag, (\Sigma_a)_{a\in Ag}, (v_a)_{a\in Ag})$*, where (i)* Ag *is finite set of agents, (ii)* Σ_a *is a set of actions for agent* $a \in Ag$*, and (iii)* $v_a : \prod_{a\in Ag}\Sigma_a \to \mathbb{Z}$ *with* $v_a(\alpha)$ *defining the outcome for agent* a *of strategy profile* $\alpha \in \Sigma = \prod_{a\in Ag}\Sigma_a$.

Then, the set of (pure) Nash Equilibria (NE) for Γ *is given as*

$$NE(\Gamma) = \{\alpha\in\Sigma \mid \textit{for all } \beta_a\in\Sigma_a, v_a(\alpha_{-a},\beta_a) \leq v_a(\alpha)\}$$

To such a game Γ *we associate the following CGS* $\mathbf{G}(\Gamma) = \langle Ag, AP, S, s_0, \{Act_a\}_{a\in Ag}, \tau, L\rangle$ *where*

- $S = \{s_0\} \cup \{(z_a)_{a\in Ag} \in \mathbb{Z}^{Ag} \mid$ *for some* $\alpha \in \Sigma$*, for all* $a \in Ag, v_a(\alpha) = z_a\}$.
- $Act_a = \Sigma_a \cup \{nop\}$.
- *For all* $\alpha \in \Sigma$, $\tau(s_0,\alpha) = (v_a(\alpha))_{a\in Ag}$ *and* $\tau(v,\alpha) = \emptyset$. *Also,* $\tau(s_0, nop) = \emptyset$*, and* $\tau(v, nop) = v$ *for all* $v \in S$, $v \neq s_0$.
- $AP = \bigcup AP_a$ *with* $AP_a = \{p_{aj} \mid 2^j \leq K_\Gamma\}$*, where* K_Γ *is the maximal integer representing a state in* S.
- *For a given boolean combination* $\eta = \bigwedge_j p_{aj}^{n_j}$ *over* AP_a *with* $n_j \in \{-1,1\}$*, we denote as* $int(\eta)$ *the integer value with binary encoding* $p_{a0}^{n_0}\dots p_{ak}^{n_k}$ *with* $k = \log K_\Gamma$ *being the maximal index of atoms in* AP_a*. Conversely, for each integer* z *we denote as* $enc(z)$ *the boolean combination corresponding to the binary encoding of* $z\mathbb{Z}$ *over* $k+1$ *bits. Then,* $L(s_0) = \emptyset$ *and* $L(z) = \{p_{aj} \mid p_{aj}$ *occurs with positive power in* $enc(v)\}$.

Consider now the following formula in SSL^-*:*

$$\varphi_\Gamma(\vec{y}) ::= \bigwedge_{z\leq K_\Gamma} enc(z) \to \bigwedge_{a\in Ag} [\![y']\!](y',a) \bigvee_{z'\leq z} enc(z')$$

We immediately state the following remark:

Remark 11. *An assignment* $\chi : \vec{y} \mapsto (\sigma_a)_{a\in Ag}$ *satisfies formula* $\varphi_\Gamma(\vec{y})$ *in state* s_0 *iff the strategy profile* $(\sigma_a(s_0))_{a\in Ag}$ *is a NE for* Γ*, i.e.,*

$$NE(\Gamma) = \{(\sigma_a(s_0))_{a\in Ag} \mid (\mathbf{G}(\Gamma), s_0, \chi) \models \varphi_\Gamma(\vec{y})\}$$

In particular, fragment SSL^- *is expressive enough to describe NE. Incidentally, we observe that formulas* $\varphi_\Gamma(\vec{y})$ *for NE live in the intersection between* SSL^- *and the boolean-goal fragment* $\mathsf{SL[BG]}$*. Note that existence of NE can be encoded as* SSL^- *formulas quadratic in the size of the normal-form game, if integers are represented in unary.*

2.3 A Comparison between SSL^- and ATL^*_{sc}

Above we said that fragments SSL, SSL^-, and $\mathsf{SSL}^-[\mathsf{1G}]$ have not been considered in the literature to our knowledge. Actually, in this section we compare SSL^- with the logic ATL^*_{sc} (Brihaye et al. 2009; Costa, Laroussinie, and Markey 2010; Laroussinie and Markey 2015), which is an extension of the well-known Alternating-time Temporal Logic ATL^* with *strategy contexts*. In particular, we prove that SSL^- and ATL^*_{sc} are equally expressive. We follow (Laroussinie and Markey 2015) in the presentation of ATL^*_{sc} and refer to it for a detailed discussion of the logic and its motivations.

Definition 12 (ATL^*_{sc})**.** *The state formulas* ϕ *and path formulas* ψ *in* ATL^*_{sc} *are defined as follows:*

$$\begin{aligned}\phi &::= p \mid \neg\phi \mid \phi\wedge\phi \mid \langle\cdot A\cdot\rangle\psi \mid (|A|)\phi\\ \psi &::= \phi \mid \neg\psi \mid \psi\wedge\psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi\end{aligned}$$

where $p \in AP$ *and* $A \subseteq Ag$.

ATL^*_{sc} *is the set of all and only state formulas.*

We recall from (Laroussinie and Markey 2015) that formula $\langle\cdot A\cdot\rangle\psi$ intuitively means that coalition A has a joint strategy such that, *given the joint strategy currently played by the complement coalition* $\overline{A}$, the outcome satisfies ψ. On the other hand, formula $(|A|)\phi$ is read as "all outcomes obtained by dropping the joint strategy currently played by coalition A satisfy ϕ".

To provide a formal semantics to ATL^*_{sc}, given a joint strategy σ_A for coalition $A \subseteq Ag$, we define the set $\mathsf{out}(h,\sigma_A)$ of the *outcomes of* σ_A *from history* h as the set of computations λ that start with h and are consistent with σ_A. Given joint strategies σ_A and σ_B for possibly overlapping coalitions A and B, the *updated profile* $\sigma_A \circ \sigma_B$ coincides with σ_B on agents in $B\setminus A$ and with σ_A on A; while $(\sigma_B)_{\setminus A}$ denotes the restriction of σ_B to coalition $B \setminus A$.

Definition 13 (Satisfaction)**.** *Let* $\mathbf{G}$ *be a CGS,* λ *a computation,* $n \in \mathbb{N}$*, and* σ_B *a joint strategy for coalition* B*. We define the satisfaction relation* $\models$ *by induction as follows (clauses for boolean operators are immediate, and thus omitted):*

$(\mathbf{G},\lambda,n) \models_{\sigma_B} p$ *iff* $p \in L(\lambda_n)$
$(\mathbf{G},\lambda,n) \models_{\sigma_B} \mathsf{X}\,\psi$ *iff* $(\mathbf{G},\lambda,n+1) \models_{\sigma_B} \psi$
$(\mathbf{G},\lambda,n) \models_{\sigma_B} \psi\,\mathsf{U}\,\psi'$ *iff for some* $i \geq n$, $(\mathbf{G},\lambda,i) \models \psi'$*, and for all* j, $0 \leq j < i$ *implies* $(\mathbf{G},\lambda,j) \models \psi$
$(\mathbf{G},\lambda,n) \models_{\sigma_B} \langle\cdot A\cdot\rangle\psi$ *iff for some joint strategy* σ_A*, for every* $\lambda' \in \mathsf{out}(\lambda_{\leq n}, \sigma_A \circ \sigma_B)$, $(\mathbf{G},\lambda',n) \models_{\sigma_A\circ\sigma_B} \psi$
$(\mathbf{G},\lambda,n) \models_{\sigma_B} (|A|)\phi$ *iff* $(\mathbf{G},\lambda,n) \models_{(\sigma_B)_{\setminus A}} \phi$

The main result of this section states that ATL^*_{sc} and SSL^- are actually equivalent formalisms. To prove this, we introduce translations tr_1 and tr_2.

Definition 14. *Consider the following translation functions $tr_1 : \mathsf{SSL}^- \to \mathsf{ATL}^*_{sc}$ and $tr_2 : \mathsf{ATL}^*_{sc} \to \mathsf{SSL}^-$, that are the identity function on atoms (i.e., $tr_1(p) = tr_2(p) = p$, for $p \in AP$), commute with boolean operators and* LTL *modalities, and for $A = \{a_1, \dots, a_k\}$,*

$$\begin{aligned}
\tau_1(\langle\langle x\rangle\rangle(x,a)\varphi) &= \langle\cdot\{a\}\cdot\rangle\tau_1(\varphi)\\
\tau_2(\langle\cdot A\cdot\rangle\varphi) &= \langle\langle x_1\rangle\rangle(x_1,a_1)\dots\langle\langle x_k\rangle\rangle(x_k,a_k)\tau_2(\varphi)\\
\tau_2((|A|)\varphi) &= [\![x_1]\!](x_1,a_1)\dots[\![x_k]\!](x_k,a_k)\tau_2(\varphi)
\end{aligned}$$

Hereafter, given an assignment χ, σ_χ is the joint strategy that assigns strategy $\chi(a)$ to each agent $a \in Ag$. Also, given a joint strategy σ_A, χ_{σ_A} is any assignment that coincides with σ_A on the agents in A.

We can now state the main equivalence result in this section, a proof can be found in the accompanying document.

Proposition 15. *Let $\mathbf{G}$ be a CGS.*

1. *For every history h, assignment χ, and SSL^- formula φ,*

$$(\mathbf{G}, h, \chi) \models \varphi \quad \textit{iff} \quad (\mathbf{G}, \lambda(h,\chi), |h|) \models_{\sigma_\chi} tr_1(\varphi)$$

2. *For every joint strategy σ_B, computation $\lambda \in \mathsf{out}(\lambda_{\leq n}), \sigma_B)$, ATL^*_{sc} formula ϕ, and assignment χ_{σ_B},*

$$(\mathbf{G}, \lambda, n) \models_{\sigma_B} \phi \quad \textit{iff} \quad (\mathbf{G}, \lambda_{\leq n}, \chi_{\sigma_B}) \models tr_2(\phi)$$

By Prop. 15 the logics SSL^- and ATL^*_{sc} are essentially equivalent, and since both translations tr_1 and tr_2 are polynomial functions, the complexity results available for ATL^*_{sc} apply to SSL^- as well. On the other hand, the peculiar syntax of SSL^- makes the relationship with SL apparent, and it prompts the question for languages between SSL^- and SL, including SSL. In Sec. 5 we will see that SSL is strictly more expressive than SSL^-, and therefore also than ATL^*_{sc}.

3 Bisimulations for SL[1G] and SSL$^-$[1G]

In this section we first introduce bisimulations that preserve the fragments SL[1G] and SSL^-[1G]. In particular, we show that such bisimulations enjoy the Hennessy-Milner property. In Sec. 5 we will apply them to the analysis of the expressive power of these two fragments of SL.

3.1 Bisimulation Relations

We first define bisimulations for SSL^-[1G] by adapting a similar notion appearing in (Mogavero 2013, Def. 3.3). Hereafter $\mathbf{G} = \langle Ag, AP, S, s_0, \{Act_a\}_{a\in Ag}, \tau, L\rangle$ and $\mathbf{G}' = \langle Ag, AP, S', s_0', \{Act'_a\}_{a\in Ag}, \tau', L'\rangle$ are CGS defined on the same sets Ag and AP of agents and atoms.

Definition 16 (SSL^-[1G]-bisimulation)**.** *Let $\mathbf{G}$ and $\mathbf{G}'$ be CGS defined on the same sets Ag and AP of agents and atoms. Then, $\mathbf{G}$ and $\mathbf{G}'$ are* SSL^-[1G]-bisimilar *iff there exists one relation $\sim \subseteq S \times S'$ between states, called* bisimulation relation*, and a family of functions $g = (g_a)_{a\in Ag}$ with $g_a : \sim \to 2^{Act_a \times Act'_a}$, called* bisimulation function*, such that*

1. *$s_0 \sim s'_0$;*
2. *for all $s \in S$, $s' \in S'$, if $s \sim s'$ then*
   (a) *$L(s) = L'(s')$;*
   (b) *for all $a \in Ag$, $\alpha \in Act_a$, there is $\alpha' \in Act'_a$ such that $(\alpha, \alpha') \in g_a(s, s')$;*
   (c) *for all $a \in Ag$, $\alpha' \in Act'_a$, there is $\alpha \in Act_A$ such that $(\alpha, \alpha') \in g_a(s, s')$;*
   (d) *for all $\vec{\alpha} \in ACT$, $\vec{\alpha}' \in ACT'$, if $(\vec{\alpha}, \vec{\alpha}') \in \hat{g}(s, s')$ then $\tau(s, \vec{\alpha}) \sim \tau'(s', \vec{\alpha}')$, where $\hat{g} : \sim \to 2^{ACT \times ACT'}$ is the pointwise lifting of g to joint actions, defined by $\hat{(g)}(s, s') = \{(\alpha, \alpha') \mid \alpha \in ACT, \alpha' \in Act'$ and $\forall a \in Ag, (\alpha_a, \alpha'_a) \in g_a(s, s')\}$.*

Hereafter we extend relation $\sim$ to histories by requiring that for every $h \in S^+, h' \in S'^+$, $h \sim h'$ if $|h| = |h'|$ and $h_i \sim h'_i$ for all $i \leq |h|$. Further, given assignments χ, and χ', we write $\hat{\hat{g}}(\chi, \chi')$ iff for every $a \in Ag$, $h \in S^+$, $h' \in S'^+$, $h \sim h'$ implies $(\chi(a)(h), \chi'(a)(h')) \in g_a(last(h), last(h'))$. That is, assignments χ and χ' are related by $\hat{\hat{g}}$ iff they return g_a-related actions on bisimilar histories, for every agent $a \in Ag$.

The following auxiliary lemma is key to prove the main preservation result.

Lemma 17. *For all histories $h \in S^+$, $h' \in S'^+$, and assignments χ, χ', if $h \sim h'$ and $\hat{\hat{g}}(\chi, \chi')$, then for every $i \geq |h|$, $\lambda(h, \chi)_{\leq i} \sim \lambda(h', \chi')_{\leq i}$.*

By using Lemma 17 we are able to prove the main preservation result of this section.

Theorem 18. *Given CGS $\mathbf{G}$ and $\mathbf{G}'$, histories $h \in S^+$, $h' \in S'^+$ and assignments χ, χ', if $h \sim h'$ and $\hat{\hat{g}}(\chi, \chi')$, then for every formula φ in* SSL^-[1G],

$$(\mathbf{G}', h', \chi') \models \varphi \quad \textit{iff} \quad (\mathbf{G}, h, \chi) \models \varphi$$

By Theorem 18 the notion of bisimulation in Def. 16 indeed preserves the interpretation of formulas in SSL^-[1G]. To preserve the whole of SL[1G] we need to consider the following strengthening of Def. 16.

Definition 19 (SL[1G]-bisimulation)**.** *Let $\mathbf{G}$ and $\mathbf{G}'$ be CGS defined on the same sets Ag and AP of agents and atoms. Then, $\mathbf{G}$ and $\mathbf{G}'$ are* SL[1G]-bisimilar *iff there are a bisimulation relation $\sim \subseteq S \times S'$ between states, and for every agent $a \in Ag$, a bisimulation function $g_a : \sim \to 2^{Act_a \times Act'_a}$ such that conditions 1, 2.(a), 2.(d) in Def 16 hold, while conditions 2.(b) and 2.(c) are substituted by the following:*

2. *for all $s \in S$, $s' \in S'$, if $s \sim s'$ then*
   (b') *for all $A \subseteq Ag$, $\alpha \in \bigcap_{a\in A} Act_a$, there is $\alpha' \in \bigcap_{a\in A} Act'_a$ such that $(\alpha, \alpha') \in g_a(s, s')$ for every $a \in A$;*
   (c') *for all $A \subseteq Ag$, $\alpha' \in \bigcap_{a\in A} Act'_a$, there is $\alpha \in \bigcap_{a\in A} Act_A$ such that $(\alpha, \alpha') \in g_a(s, s')$ for every $a \in A$.*

The stronger conditions 2.(b') and 2.(c') in Def. 19 reflect the fact that, differently from SSL^-[1G], the syntax of SL[1G] allows agents to share strategies. By using this stronger notion of bisimulation we are able to prove the preservation of formulas in SL[1G]. In particular, 2.(b') and 2.(c') are used to deal with formulas of type $\varphi = \wp\flat\varphi'$.

Theorem 20. *Given CGS* $\mathbf{G}$ *and* $\mathbf{G}'$*, histories* $h \in S^+$, $h' \in S'^+$*, and assignments* χ, χ'*, if* $h \sim h'$ *and* $\hat{g}(\chi, \chi')$*, then for every formula* φ *in* SL[1G],

$$(\mathbf{G}, h, \chi) \models \varphi \quad \textit{iff} \quad (\mathbf{G}', h', \chi') \models \varphi$$

By Theorem 18 and 20, SSL$^-$[1G]- and SL[1G]-bisimulations are sufficient to preserve the interpretation of formulas in SSL$^-$[1G] and SL[1G] respectively. In the following section we prove that they are also necessary.

3.2 Bisimulation Games and the HM Property

In this section we define bisimulation games for both SL[1G] and SSL$^-$[1G], we prove them equivalent to the bisimulation relations in Def. 16 and 19, and finally we show that both notions of bisimulation enjoy the Hennessy-Milner property.

Definition 21 (Bisimulation Game for SSL$^-$[1G])**.** *Given CGS* $\mathbf{G}$ *and* $\mathbf{G}'$*, defined on the same sets Ag of agents and AP of atoms, we define the bisimulation game* $G(\mathbf{G}, \mathbf{G}')$ *as a game between two players,* Spoiler *S and* Duplicator *D, whose initial position is* (s_0, s_0')*, and in every position* (s, s') *of the game Spoiler and Duplicator play as follows:*

1. *Check whether* $L(s) = L'(s')$*. If that is the case, the game proceeds; otherwise, the game terminates and S wins.*
2. *For some agent* $a \in Ag$*, S picks either an action* $\alpha_a \in Act_a$ *or an action* $\alpha'_a \in Act'_a$*.*
3. *If S picked an action in* Act_a*, then D has to reply with an action* $\beta'_a \in Act'_a$*; otherwise, D replies with an action* $\beta_a \in Act_a$
4. *S and D continue to pick actions for all agents in Ag.*
5. *Finally, we end up with joint actions* $\langle \alpha_1, \ldots, \alpha_{|Ag|} \rangle$ *in* $\mathbf{G}$ *and* $\langle \alpha'_1, \ldots, \alpha'_{|Ag|} \rangle$ *in* $\mathbf{G}'$*. Then,* $(\tau(s, \langle \alpha_1, \ldots, \alpha_{|Ag|} \rangle), \tau'(s', \langle \alpha'_1, \ldots, \alpha'_{|Ag|} \rangle))$ *is the new position of the game.*

If the game does not terminate, then D wins. Otherwise, the game reaches a position from which D cannot choose as required, and S wins.

Next we prove that bisimulation relations and games are equivalent characterisations of CGS.

Theorem 22. *CGS* $\mathbf{G}$ *and* $\mathbf{G}'$ *are bisimilar iff Duplicator can win the bisimulation game* $G(\mathbf{G}, \mathbf{G}')$*.*

Proof. $\Rightarrow$ If $\mathbf{G}$ and $\mathbf{G}'$ are bisimilar, then every position (s, s') visited during the game is such that $s \sim s'$. This is true for (s_0, s_0'). Further, at every position (s, s') D can match an action α_a, for some agent a, with action α'_a such that either $(\alpha_a, \alpha'_a) \in g_a(s, s')$ or $(\alpha'_a, \alpha_a) \in g_a(s, s')$. Such action exists by $2.(b)$ and $2.(c)$. Then, we obtain that the new position $(\tau(s, \vec{\alpha}), \tau'(s', \vec{\alpha}'))$ of the game also satisfies $\tau(s, \vec{\alpha}) \sim \tau'(s', \vec{\alpha}')$ by $2.(d)$. Hence, D can win the bisimulation game $G(\mathbf{G}, \mathbf{G}')$.

$\Leftarrow$ We define a bisimulation relation $\sim \subseteq S \times S'$ by setting $s \sim s'$ iff "(s, s') is a winning position for D", and for every agent $a \in Ag$, g_a is the function such that $(\alpha, \alpha') \in g_a(s, s')$ iff "in state (s, s') D can reply with action α (resp. α') to action α' (resp. α)". Clearly, all conditions 1 and $2.(a)$-(d) on bisimulation relations and functions are satisfied (otherwise (s_0, s_0') would not be a winning position for D), and therefore $\mathbf{G}$ and $\mathbf{G}'$ are bisimilar. □

As a consequence of Theorem 22, we obtain the converse of Theorem 18.

Corollary 23. *Bisimulations for* SSL$^-$[1G] *enjoy the Hennessy-Milner property. That is, if CGS* $\mathbf{G}$ *and* $\mathbf{G}'$ *are logically equivalent w.r.t.* SSL$^-$[1G]*, then they are* SSL$^-$[1G]*-bisimilar.*

Proof. By Theorem 22 if $\mathbf{G}$ and $\mathbf{G}'$ are not bisimilar, then S can win the game in a finite number of steps. By using this finite play and reasoning along the lines of (Alur et al. 1998, Theorem 6) we can construct an SSL$^-$[1G] formula that is true in $\mathbf{G}$ but false in $\mathbf{G}'$. Hence, the two CGS are not logically equivalent. □

Furthermore, we observe that bisimulation games can be defined for SL[1G] as well. These are obtained by modifying points (2) and (3) in Def. 21 as follows:

2′. For some coalition $A \subseteq Ag$, S picks either an action $\alpha_A \in \bigcap_{a \in A} Act_a$ or an action $\alpha'_A \in \bigcap_{a \in A} Act'_a$.

3′. If S picked an action in $\alpha_A \in \bigcap_{a \in A} Act_a$, then D has to reply with an action $\beta'_A \in \bigcap_{a \in A} Act'_a$; otherwise, D picks an action $\beta_A \in \bigcap_{a \in A} Act_a$

Similarly to Theorem 22 and Corollary 23, we can then prove that the CGS $\mathbf{G}$ and $\mathbf{G}'$ are bisimilar for SL[1G] iff Duplicator wins the bisimulation game $G(\mathbf{G}, \mathbf{G}')$. Moreover, bisimulations for SL[1G] also enjoy the Hennessy-Milner property.

4 Bisimulations for SSL and SSL$^-$

Here we put forward notions of bisimulation for SSL and SSL$^-$, then prove that they indeed preserve the interpretation of formulas in the relevant fragment. Differently from the previous section, these bisimulations do not enjoy the Hennessy-Milner property.

4.1 Bisimulation Relations

We start with the bisimulations for SSL$^-$. Hereafter, given a bisimulation relation $\sim \subseteq S \times S'$, let $\sim(s) = \{s' \in S \mid s \sim s'\}$ and $\sim^{-1}(s') = \{s \in S \mid s \sim s'\}$.

Definition 24 (SSL$^-$-bisimulation)**.** *Let* $\mathbf{G}$ *and* $\mathbf{G}'$ *be CGS defined on the same sets* Ag *and* AP *of agents and atoms. Then,* $\mathbf{G}$ *and* $\mathbf{G}'$ *are* SSL$^-$-bisimilar *iff there are a bisimulation relation* $\sim \subseteq S \times S'$ *between states, and for every agent* $a \in Ag$*, a bisimulation function* $g_a : \sim \rightarrow 2^{Act \times Act'}$ *such that conditions* 1, 2.(a), 2.(d) *in Def 16 hold, while conditions* 2.(b) *and* 2.(c) *are substituted by the following:*

2. *for all* $s \in S$ *and* $s' \in S'$*, if* $s \sim s'$ *then*

(b'') *For all* $a \in Ag$*, for all tuples* $\langle \alpha_1, \ldots, \alpha_{|\sim^{-1}(s')|} \rangle \in Act_a^{|\sim^{-1}(s')|}$*, there is* $\alpha' \in Act'_a$ *such that* $(\alpha_i, \alpha') \in g_a(s_i, s')$ *for every* $i \leq |\sim^{-1}(s')|$*;*

(c'') *For all* $a \in Ag$*, for all tuples* $\langle \alpha'_1, \ldots, \alpha'_{|\sim(s)|} \rangle \in Act'^{|\sim(s)|}_a$*, there is* $\alpha \in Act_a$ *such that* $(\alpha, \alpha'_i) \in g_a(s, s'_i)$ *for every* $i \leq |\sim(s)|$*.*

Def. 24 differs from Def. 16 as regards conditions 2.(b) and 2.(c). In particular, we here require that if state s is bisimilar with several states $s_1, \ldots, s_n$, then for any choice of actions $\alpha_1, \ldots, \alpha_n$ in $s_1, \ldots, s_n$, there is a single action α in s mimicking such a choice. This condition is key to prove Lemma 25 below. Clearly, all CGS bisimilar according to Def. 24 are also bisimilar for Def. 16. In particular, notice that Lemma 17 still holds.

Lemma 25. *If CGS* $\mathbf{G}$ *and* $\mathbf{G}'$ *are* SSL^-*-bisimilar, then for every* a*-strategy* σ *in* $\mathbf{G}$ *there exists an* a*-strategy* σ' *in* $\mathbf{G}'$ *s.t.* $\hat{\hat{g}}(\sigma, \sigma')$*, and viceversa.*

By Lemma 25 we are able to prove the main preservation result of this section.

Theorem 26. *Given CGS* $\mathbf{G}$ *and* $\mathbf{G}'$*, histories* $h \in S^+$*,* $h' \in S'^+$*, and assignments* χ*,* χ'*, if* $h \sim h'$ *and* $\hat{g}(\chi, \chi')$*, then for every formula* φ *in* SSL^-*,*

$$(\mathbf{G}, h, \chi) \models \varphi \quad \textit{iff} \quad (\mathbf{G}', h', \chi') \models \varphi$$

By Theorem 26 the bisimulations in Def. 24 preserve the formulas in SSL^-. We can now extend this result to SSL.

Definition 27 (SSL-bisimulation)**.** *Let* $\mathbf{G}$ *and* $\mathbf{G}'$ *be CGS defined on the same sets Ag and AP of agents and atoms. Then,* $\mathbf{G}'$ *and* $\mathbf{G}$ *are* SSL-bisimilar *iff there are a bisimulation relation* $\sim \subseteq S \times S'$*, and for every agent* $a \in Ag$*, a bisimulation function* $g_a : \sim \to 2^{Act_a \times Act'_a}$ *such that conditions* 1, 2.(a), 2.(d) *in Def 16 hold, while conditions* 2.(b) *and* 2.(c) *are substituted by the following:*

2. *for all* $s \in S$*,* $s' \in S'$*, if* $s \sim s'$ *then*

(b''') *for all* $A \subseteq Ag$*, for all* $\alpha_1, \ldots, \alpha_{|\sim^{-1}(s')|} \in \bigcap_{a \in A} Act_a$*, there is* $\alpha' \in \bigcap_{a \in A} Act'_a$ *such that* $(\alpha_i, \alpha') \in g_a(s_i, s')$ *for every* $a \in A$*,* $i \leq |\sim^{-1}(s')|$*;*

(c''') *for all* $A \subseteq Ag$*, for all* $\alpha'_1, \ldots, \alpha'_{|\sim(s)|} \in \bigcap_{a \in A} Act'_a$*, there is* $\alpha \in \bigcap_{a \in A} Act_a$ *such that* $(\alpha, \alpha'_i) \in g_a(s, s'_i)$ *for every* $a \in A$*,* $i \leq |\sim(s)|$*.*

As it was the case for Def. 19, the stronger conditions 2.(b''') and 2.(c''') in Def. 27 reflect the fact that, differently from SSL^-, the syntax of SSL allows agents to share strategies. For this (stronger) notion of bisimulation we are able to prove a version of Theorem 26 extended to SSL. In particular, conditions 2.(b''') and 2.(c''') are used to deal with formulas of type $\varphi = \wp\flat\varphi'$.

Theorem 28. *Given CGS* $\mathbf{G}$ *and* $\mathbf{G}'$*, histories* $h \in S^+$*,* $h' \in S'^+$*, and assignments* χ*,* χ'*, if* $h \sim h'$ *and* $\hat{g}(\chi, \chi')$*, then for every formula* φ *in* SSL*,*

$$(\mathbf{G}, h, \chi) \models \varphi \quad \textit{iff} \quad (\mathbf{G}', h', \chi') \models \varphi$$

By Theorem 26 and 28, SSL^-- and SSL-bisimulations preserve formulas in SSL^- and SSL respectively.

4.2 Failure of the HM Property

We now prove that, differently from the bisimulations in Sec. 3, bisimulations for SSL and SSL^- do not enjoy the full Hennessy-Milner property, that is, the following holds.

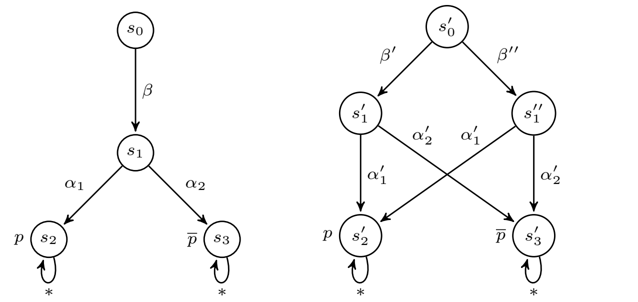

Figure 2: the CGS $\mathbf{G}$ and $\mathbf{G}'$ used in the proof of Lemma 29.

Lemma 29. *There exists CGS that are logically equivalent w.r.t.* SSL *(resp.* SSL^-*), but are not bisimilar for* SSL *(resp.* SSL^-*).*

Proof. First notice that for a single agent, the syntax of SSL (and *a fortiori* SSL^-) coincides with $\mathsf{SSL}^-[1\mathsf{G}]$. Then, consider the CGS depicted in Fig. 2 for $Ag = \{1\}$, and define relation $\sim = \{(s_0, s'_0), (s_1, s'_1), (s_1, s''_1), (s_2, s'_2), (s_3, s'_3)\}$ on $S \times S'$, and function $g_1 : \sim \to 2^{Act_1 \times Act'_1}$ such that

$$g_1(s_0, s'_0) = \{(\beta, \beta'), (\beta, \beta'')\}$$
$$g_1(s_1, s'_1) = g_1(s_1, s''_1) = \{(\alpha_1, \alpha'_1), (\alpha_2, \alpha'_2)\}$$
$$g_1(s_2, s'_2) = g_1(s_3, s'_3) = \{(*, *)\}$$

Clearly, $\sim$ is a bisimulation relation on $S \times S'$ and g_1 is a bisimulation function for agent 1 according to Def. 16. Hence, by Theorem 18 the same formulas in $\mathsf{SSL}^-[1\mathsf{G}]$ are true in s_0 and s'_0, and therefore in SSL (and SSL^-), as all these logics coincide for the single agent case.

However, CGS $\mathbf{G}$ and $\mathbf{G}'$ are not bisimilar for SSL^- (and *a fortiori* for SSL). Specifically, notice that $s_1 \sim s'_1$ and $s_1 \sim s''_1$. But there is no $\alpha \in Act_1$ such that both $(\alpha, \alpha'_1) \in g_1(s_1, s'_1)$ and $(\alpha, \alpha'_2) \in g_1(s_1, s''_1)$. ☐

Remark 30. *Note that this negative result is not related with Theorem 3 from (Gutierrez et al. 2017), which is about labeled bisimilarity. To see this, note that the two CGS in Fig. 2 are labeled bisimilar in the sense of (Gutierrez et al. 2017).*

5 Expressivity and Verification

We now make use of the bisimulations introduced in Sec. 3 and 4 to analyse the expressive power of the various fragments of SL. We first introduce a notion of being *at least as expressive as* for logics.

Definition 31 (at least as expressive as)**.** *A logic* L *is* at least as expressive as *a logic* L'*, or* $L' \leq L$*, iff for every* $\varphi' \in L'$ *there exists* $\varphi \in L$ *such that* φ *and* φ' *are logically equivalent.*

Logics L and L' are equally expressive, or $L \equiv L'$, iff $L \leq L'$ and $L' \leq L$. Finally, L is *strictly more expressive than* L', or $L' < L$, iff $L' \leq L$ but $L' \not\equiv L$. Clearly, $\leq$ and $<$ are a partial and strict order respectively (hence the notation).

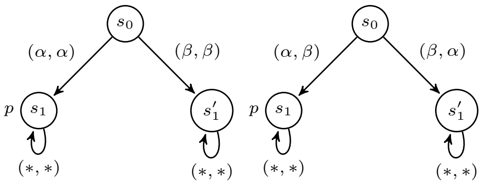

Figure 3: the CGS $\mathbf{G}_3$ and $\mathbf{G}_4$ in the proof of Lemma 32.1.

5.1 Inclusions

Hereafter we show that for all fragments of SL but the nested- and boolean-goal fragments, relation $<$ matches the syntactic inclusions in Fig. 1, as reported in Fig. 4. We do not state results for the nested- and boolean-goal fragments, as these are outside the scope of the present paper.

We first show that SL[1G] is strictly more expressive than SSL^-[1G], and $\mathsf{SSL}^- < \mathsf{SSL} < \mathsf{SL}$.

Lemma 32. *The following holds:*

1. $\mathsf{SSL}^-[1\mathsf{G}] < \mathsf{SL}[1\mathsf{G}]$
2. $\mathsf{SSL} < \mathsf{SL}$
3. $\mathsf{SSL}^- < \mathsf{SSL}$

Proof. (1) Clearly, $\mathsf{SSL}^-[1\mathsf{G}] \leq \mathsf{SL}[1\mathsf{G}]$, so we prove that $\mathsf{SL}[1\mathsf{G}] \not\leq \mathsf{SSL}^-[1\mathsf{G}]$. Consider the CGS $\mathbf{G}_3 = \langle Ag, AP, S, s_0, \{Act_a\}_{a \in Ag}, \tau_3, L\rangle$ and $\mathbf{G}_4 = \langle Ag, AP, S, s_0, \{Act_a\}_{a \in Ag}, \tau_4, L\rangle$ in Fig. 3, with $Ag = \{1, 2\}$, $AP = \{p\}$, $Act_0 = Act_1 = \{\alpha, \beta\}$, $S = \{s_0, s_1, s_1'\}$, and $L(s_1) = \{p\}$, $L(s) = \emptyset$ for all other $s \in S$.

We can see that $\mathbf{G}_3$ and $\mathbf{G}_4$ are SSL^-[1G]-bisimilar by assuming $\sim \{(s, s) \mid s \in S\}$, $g_1(s, s) = \{(c, c) \mid c \in Act_1 = Act_2\}$, for every $s \in S$, and $g_2(s_0, s_0) = \{(\alpha, \beta), (\beta, \alpha)\}$. Hence, by Theorem 20, $\mathbf{G}_3$ and $\mathbf{G}_4$ satisfy the same formulas in SSL^-[1G]. However, $\mathbf{G}_3$ satisfies the SL[1G]-formula $\psi = \langle\langle x\rangle\rangle(x, 1)(x, 2)\, \mathsf{X}\, p$, while $\mathbf{G}_4$ does not. Thus, ψ has no equivalent in SSL^-[1G].

(2) The inclusion $\mathsf{SSL} \leq \mathsf{SL}$ is immediate. To prove that $\mathsf{SL} \not\leq \mathsf{SSL}$, consider the following formula in SL:

$$\phi = \langle\langle x\rangle\rangle\big(\langle\langle y\rangle\rangle(y, a)\, \mathsf{X}(x, a)\, \mathsf{X}\, p \wedge \langle\langle y\rangle\rangle(y, a)\, \mathsf{X}(x, a)\, \mathsf{X}\, \neg p\big)$$

Then, consider again the single-agent CGS in Fig. 2, which were proved to be SSL^-[1G]-isomorphic in the proof of Lemma 29. In particular, the same formulas in SSL are true in s_0 and s_0', as SSL^-[1G] and SSL coincide syntactically for the single-agent case. However, we can check that $(\mathbf{G}, s_0) \not\models \phi$, but $(\mathbf{G}', s_0') \models \phi$. As to the latter, substitute variable x with strategy σ such that $\sigma(s_1') = \alpha_1'$ et $\sigma(s_1'') = \alpha_2'$. As a result, ϕ has no equivalent formula in SSL.

(3) Again, $\mathsf{SSL}^- \leq \mathsf{SSL}$. As regards $\mathsf{SSL} \not\leq \mathsf{SSL}^-$, observe that the CGS $\mathbf{G}_3$ and $\mathbf{G}_4$ in Fig. 3 are SSL^--bisimilar as well, but as remarked above, they satisfy different formulas in SL[1G], and therefore in SSL. ☐

Next we state that logics SL[1G] and SSL^- are incomparable. A formal proof can be found in the accompanying document.

SL SSL SSL^-	Tower-c	(Mogavero et al. 2014) (Laroussinie and Markey 2015)
SL[1G] SSL^-[1G]	2EXPTIME-c	(Mogavero et al. 2014)

Table 1: Complexity results for the model checking problem.

Lemma 33. *The following holds:*

1. $\mathsf{SL}[1\mathsf{G}] \not\leq \mathsf{SSL}^-$
2. $\mathsf{SSL}^- \not\leq \mathsf{SL}[1\mathsf{G}]$

The results of this section allow us to answer two fundamental questions left open for years about Strategy Logic. Firstly, the main difference between the two versions of Strategy Logic introduced in (Chatterjee, Henzinger, and Piterman 2010) and (Mogavero et al. 2014) respectively is that the former does not allow the sharing of strategies, while the latter instead does. A natural question is: does this feature really add expressive power to SL? here we answered positively this question as regards fragments SL[1G] and $\mathsf{SSL}^- = \mathsf{ATL}^*_{sc}$.

Secondly, ATL^*_{sc} and SL share similar capabilities (e.g., they can both express Nash Equilibria) and complexity results regarding the model checking and the satisfiability problems (Laroussinie and Markey 2015). For a long time there has been a discussion as to whether SL is more expressive than ATL^*_{sc}. The results in this section answer positively also this second question.

5.2 The Model Checking Problem

We now analyse the model checking problem in the light of the expressivity results in the previous section. First of all, we state formally this problem for a logic L.

Definition 34 (Model-checking for L)**.** *Given a CGS* $\mathbf{G}$ *and a formula* ϕ *in L, determine whether* $\mathbf{G} \models \phi$*.*

We recall that model checking both SL and ATL^*_{sc} is Tower-complete (Mogavero et al. 2014; Laroussinie and Markey 2015). Given the (linear) translations between ATL^*_{sc} and SSL^- in Sec. 2.3, it follows that also the model checking problems for SSL^- and SSL are Tower-complete.

Further, in (Mogavero et al. 2014) the verification of SL[1G] is shown to be 2EXPTIME-complete, and above we remaked that SSL^-[1G] is at least as expressive as ATL^*, since the strategic operator $\langle\langle A\rangle\rangle$ can be translated as $\langle\langle \vec{x}\rangle\rangle(\vec{x}, A)[\![\vec{y}]\!](\vec{y}, Ag \setminus A)$. As a consequence, model checking SSL^-[1G] is 2EXPTIME-complete as well (Alur, Henzinger, and Kupferman 2002).

We summarize these complexity results for the model checking problem in Table 1. As a consequence, we obtain that there is no truth-preserving translation from SSL^- to SSL^-[1G], neither from SSL to SL[1G]. Hence, we have the following immediate corollary on expressivity:

Corollary 35. $\mathsf{SSL}^-[1\mathsf{G}] < \mathsf{SSL}^-$ *and* $\mathsf{SL}[1\mathsf{G}] < \mathsf{SSL}$

Finally, we summarize the expressity results for the various fragments of SL in Fig. 4. Notice that the (strict) or-

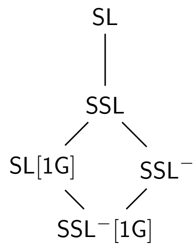

Figure 4: the partial order $<$ for fragments of SL.

der between fragments mimicks the syntactic inclusions in Fig. 1.

5.3 Elimination of Dominated Strategies

We conclude this section by providing an application of our model-theoretic results on bisimulations to a game-theoretic context. Hereafter we prove that the elimination of strictly dominated strategies generates SSL$^-$-bisimilar game structures. Since Nash Equilibria are expressible in SSL$^-$ (Example 10), we obtain a new, model-theoretic proof of the well-known result on the preservation of Nash Equilibria under the elimination of strictly dominated strategies (Osborne 2004, Lemma 60.1). We fist show that payoffs in games in normal form can be encoded in a "canonical" representation.

Recall that, given a game in normal form $\Gamma = (Ag, (\Sigma_a)_{a \in Ag}, (v_a)_{a \in Ag})$ and, for some agent $a \in Ag$, two strategies $\alpha_a, \beta_a \in \Sigma_a$, we say that α_a is *dominated by* β_a if for any strategy profile $\gamma \in \prod_{a \in Ag} \Sigma_a$, with $\gamma_a = \alpha_a$, we have that $v_a(\gamma) < v_a(\gamma_{-a}, \beta_a)$, where (γ_{-a}, β_a) is the strategy profile obtained by substituting α_a with β_a in γ.

Lemma 36. *Let $\Gamma = (Ag, (\Sigma_a)_{a \in Ag}, (v_a)_{a \in Ag})$ be a game in normal form, where every v_a is strictly positive and there are actions β_a, γ_a such that β_a is strictly dominated by γ_a. Then, consider the game in normal form $\Gamma' = (Ag, (\Sigma_a)_{a \in Ag}, (v'_a)_{a \in Ag})$, where $v'_a(\alpha) = 2 \cdot v_a(\alpha) + 1$ for every $\alpha \in \Sigma = \prod_{a \in Ag} \Sigma_a$ such that $\alpha_a \neq \beta_a$, and $v'_a(\alpha) = 2 \cdot v_a(\alpha)$ for $\alpha_a = \beta_a$. Then, the Nash Equilibria of both games Γ and Γ' are the same, i.e., $NE(\Gamma) = NE(\Gamma')$.*

The main result of this section is the following:

Proposition 37. *Consider game Γ', obtained from game Γ as per Lemma 36, and the corresponding CGS $\mathbf{G}(\Gamma')$ defined as in Example 10. Define further game $\mathbf{G}'' = \langle Ag, AP, S'', s_0, \{Act''_a\}_{a \in Ag}, \tau'', L'' \rangle$ with $S'' = S \setminus \{v \in S \mid v \text{ is odd}\}$, $Act''_b = \Sigma_b \cup \{nop\}$ for every $b \neq a$, and $Act''_a = (\Sigma_a \cup \{nop\}) \setminus \{\beta_a\}$; and let τ'', L'' be the restrictions of τ, L on $\mathbf{G}''$.*

Now consider the tuple $(\sim, (g_a)_{a \in Ag})$ defined as follows:

- *$s_0 \sim s_0$, $v \sim v$ for each $v \in S \cap S''$, and $2v + 1 \sim 2v$ for each integer v such that $2v, 2v + 1 \in S''$.*
- *For every $s \in S$, $b \in Ag$, $g_b(s, s) = diag(Act_b)$, i.e., the identity relation on ACT_b; and $g_a(2v + 1, 2v) = \{(\beta_a, \beta_a), (\gamma_a, \beta_a)\}$.*
- *$AP'' = \{p_{bj} \mid b = a \text{ implies } j \geq 1\}$.*

Then, $(\sim, (g_a)_{a \in Ag})$ is an SSL$^-$-bisimulation over the set AP'' of atoms.

Intuitively, Prop. 37 states that, in the CGS $\mathbf{G}(\Gamma')$ corresponding to the modified game Γ', the dominated action γ_a can be removed by identifying it with β_a, and redirecting each transition which leaves s_0 with a joint action α, for $\alpha_a = \gamma_a$, to the state representing the value $v + 1$, where $v = v_a(\alpha)$. This is consistent, as in Γ' we have $v_a(\alpha_{-a}, \beta_a) = v + 1$ by construction.

The proof works by a simple verification of the conditions on SSL$^-$-bisimulations in Definition 24. We here provide a sketch. Note that the bisimulation relation $\sim$ is indeed the identity, except for the case of tuples $(2v + 1, 2v)$ of states, which corresponds to the outcomes of tuples of actions containing β_a, resp. γ_a. By construction, these tuples $(2v + 1, 2v)$ are identically labeled over AP'', since the least significant bit in the base 2 expansion of $2v$, resp. $2v + 1$, is ignored. Hence, all the other bits of the two base 2 expansions are identical. Property *2.(d)* only needs checking for the case (β_a, γ_a), which holds since by construction, $\tau(s_0, (\alpha_{-a}, \beta_a)) = 2v + 1 \sim 2v = \tau(s_0, (\alpha_{-a}, \gamma_a))$. Condition *2.(b")* holds trivially, and condition *2.(c")* only needs to be checked for tuple (β_a, γ_a), which is the only case when some $\sim^{-1}(s')$ is not a singleton. For such cases *2.(c")* holds since we have both $(\beta_a, \gamma_a) \in g(s_0, s_0)$ and $(\gamma_a, \gamma_a) \in g(s_0, s_0)$.

As a result, $\mathbf{G}(\Gamma)$ and $\mathbf{G}''$ are bisimilar and they satisfy the same formulas in SSL$^-$. Since Nash Equilibria are expressible in SSL$^-$, in particular $NE(\mathbf{G}(\Gamma)) = NE(\mathbf{G}'')$. That is, Nash Equilibria are indeed preserved by the elimination of strictly dominated strategies (Osborne 2004, Lemma 60.1).

6 Conclusions

In this paper we contributed to the model theory of logics for strategies, which is a topic of growing interest in the community on formal verification of multi-agent systems. Specifically, we provided a unified framework for several relevant fragments of Strategy Logic, some of which have already been considered in the literature (SL[1G]), while others were introduced here for the first time (SSL, SSL$^-$, SSL$^-$[1G]). For all these fragments we defined notions of bisimulation and proved that they preserve the interpretation of formulas in the corresponding fragment. We also showed that the bisimulations for SL[1G] and SSL$^-$[1G] enjoy the Hennessy-Milner property; while for SSL and SSL$^-$ it fails. Then, we applied these theoretical results to the analysis of the expressive power of the fragments of SL, including their model checking problems. We showed that SSL$^-$ is the fragment of Strategy Logic equivalent to Alternating-time Temporal Logic with strategy contexts (ATL$^*_{sc}$). In general, we proved that the relative expressivity matches language inclusions, and that the complexity of model checking splits our family of logics in two: SSL and SSL$^-$ are Tower-complete, while SL[1G] and SSL$^-$[1G] are 2EXPTIME-complete. In particular, there seems to be a relation between the complexity of model checking and the failure of the Hennessy-Milner property. Finally, we illustrated the usefulness of our model-theoretic results by proving that the elimination of dominated strategies, a game-theoretic notion, generates SSL^--bisimilar game structures. Results along this line might point to interesting relationships between Game Theory and logics for strategies.

In future work we plan to address the issue of bisimulations for fragments of Strategy Logic between SL and SL[1G], including the nested- and boolean-goal fragments SL[NG] and SL[BG]. Also, it is of interest to refine our notions of bisimulation for SSL and SSL^-, in order to obtain the Hennessy-Milner property, or to identify fragments between SSL^- and $\mathsf{SSL}^-[1G]$ for which it holds, possibly with an elementary model checking problem. We expect that these notions of bisimulation, together with the appropriate Hennessy-Milner properties, help identifying fragments in which the existence of a Nash Equilibrium cannot be expressed. Further, we plan to study the bisimulation for SL in the context of imperfect information, in both cases of perfect and imperfect recall (Belardinelli et al. 2017; 2017; Berthon et al. 2017; Cermák et al. 2018). Finally, we aim at developing abstraction and refinement techniques for application to system verification.

Acknowledgements. F. Belardinelli acknowledges the support of ANR JCJC Project SVeDaS (ANR-16-CE40-0021), and A. Murano the support of a GNCS project 2018.

References

Ågotnes, T.; Goranko, V.; and Jamroga, W. 2007. Alternating-Time Temporal Logics with Irrevocable Strategies. In *Proc. of TARK'07*, 15–24.

Alur, R.; Henzinger, T. A.; Kupferman, O.; and Vardi, M. Y. 1998. Alternating Refinement Relations. In *Proc. of CONCUR'98*, 163–178. Springer-Verlag.

Alur, R.; Henzinger, T. A.; and Kupferman, O. 2002. Alternating-time Temporal Logic. *Journal of the ACM* 49(5):672–713.

Belardinelli, F.; Condurache, R.; Dima, C.; Jamroga, W.; and Jones, A. V. 2017. Bisimulations for Verifying Strategic Abilities with an Application to Threeballot. In *Proc. of AAMAS'17*, 1286–1295. IFAAMAS Press.

Belardinelli, F.; Lomuscio, A.; and Michaliszyn, J. 2016. Agent-Based Refinement for Predicate Abstraction of Multi-Agent Systems. In *Proc. of ECAI'16*, 286–294. IOS Press.

Belardinelli, F.; Lomuscio, A.; Murano A.; Rubin S. 2017. Verification of Multi-agent Systems with Imperfect Information and Public Actions. In *Proc. of AAMAS'17*, 1268–1276.

Belardinelli, F.; Lomuscio, A.; Murano A.; Rubin S. 2017. Verification of Broadcasting Multi-Agent Systems against an Epistemic Strategy Logic. In *Proc. of IJCAI'17*, 91–97.

Berthon, R.; Maubert, B.; Murano, A.; Rubin, S.; and Vardi, M.Y. 2017. Strategy Logic with Imperfect Information. In *Proc. of LICS'17*, 1–12. IEEE Computer Society Press.

Brihaye, T.; Costa, A. D.; Laroussinie, F.; and Markey, N. 2009. ATL with Strategy Contexts and Bounded Memory. In *Proc. of LFCS'09*, 92–106. Springer.

Cermák, P.; Lomuscio, A.; Mogavero, F.; and Murano, A. 2015. Practical Verification of Multi-Agent Systems against SLK Specifications. *Information and Computation* 261:588–614.

Cermák, P.; Lomuscio, A.; and Murano, A. 2015. Verifying and Synthesising Multi-Agent Systems against One-Goal Strategy Logic Specifications. In *Proc. of AAAI'15*, 2038–2044. AAAI Press.

Chatterjee, K.; Henzinger, T.; and Piterman, N. 2010. Strategy Logic. *Information and Computation* 208(6):677–693.

Clarke, E. M.; Grumberg, O.; Jha, S.; Lu, Y.; and Veith, H. 2000. Counterexample-Guided Abstraction Refinement. In *Proc. of CAV'00*, 154–169. Springer.

Costa, A. D.; Laroussinie, F.; and Markey, N. 2010. ATL with Strategy Contexts: Expressiveness and Model Checking. In *Proc. of FSTTCS'10*, LIPIcs 8, 120–132. Leibniz-Zentrum fuer Informatik.

Dastani, M., and Jamroga, W. 2010. Reasoning about Strategies of Multi-Agent Programs. In *Proc. of AAMAS'10*, 625–632.

Goltz, U.; Kuiper, R.; and Penczek, W. 1992. Propositional temporal logics and equivalences. In *Proc. of CONCUR'92*, 222–236.

Gutierrez, J.; Harrenstein, P.; Perelli, G.; and Wooldridge, M. 2017. Nash Equilibrium and Bisimulation Invariance. In *Proc. of CONCUR'17*, 17:1–17:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.

Kacprzak, M.; Nabialek, W.; Niewiadomski, A.; Penczek, W.; Pólrola, A.; Szreter, M.; Woźna, B.; and Zbrzezny, A. 2008. Verics 2007 - A Model Checker for Knowledge and Real-Time. *Fundamenta Informaticae* 85(1):313–328.

Laroussinie, F., and Markey, N. 2015. Augmenting atl with strategy contexts. *Information and Computation* (245):98–123.

Lomuscio, A.; Qu, H.; and Raimondi, F. 2017. MCMAS: A Model Checker for the Verification of Multi-Agent Systems. *Software Tools for Technology Transfer* 19(1):9–30.

Mogavero, F.; Murano, A.; Perelli, G.; and Vardi, M. 2012. What Makes ATL^* Decidable? A Decidable Fragment of Strategy Logic. In *Proc.of CONCUR'12*, 193–208. Springer.

Mogavero, F.; Murano, A.; Perelli, G.; and Vardi, M. 2017. Reasoning about Strategies: on the Satisfiability Problem. In Logical Methods in Computer Science, 13(1):1–37.

Mogavero, F.; Murano, A.; Perelli, G.; and Vardi, M. 2014. Reasoning about Strategies: On the Model-Checking Problem. *ACM Transactions in Computational Logic* 15(4):34:1–34:47.

Mogavero, F. 2013. *Logics in Computer Science*. Ph.D. Dissertation, Università di Napoli Federico II.

Osborne, M. 2004. *An introduction to game theory*. Oxford University Press.

Pauly, M. 2002. A Modal Logic for Coalitional Power in Games. *Journal of Logic and Computation* 12(1):149–166.

van Benthem, J. 1976. *Modal Correspondence Theory*. Ph.D. Dissertation, University of Amsterdam.