Dealing with imperfect information in Strategy Logic

Sophia Knight
LORIA - CNRS / Université de Lorraine
Nancy, France
sophia.knight@gmail.com

Bastien Maubert
LORIA - CNRS / Université de Lorraine
Nancy, France
bastien.maubert@gmail.com

We propose an extension of Strategy Logic (SL), in which one can both reason about strategizing under imperfect information and about players' knowledge. One original aspect of our approach is that we do not force strategies to be uniform, i.e. consistent with the players' information, at the semantic level; instead, one can express in the logic itself that a strategy should be uniform. To do so, we first develop a "branching-time" version of SL with perfect information, that we call BSL, in which one can quantify over the different outcomes defined by a partial assignment of strategies to the players; this contrasts with SL, where temporal operators are allowed only when all strategies are fixed, leaving only one possible play. Next, we further extend BSL by adding distributed knowledge operators, the semantics of which rely on equivalence relations on partial plays. The logic we obtain subsumes most strategic logics with imperfect information, epistemic or not.

1 Introduction

Over the past decade, investigation of logical systems for studying strategic abilities has thrived in the areas of artificial intelligence and multi-agent systems. However, there is still no satisfying logical framework to model, specify and analyze such systems. One of the proposals most studied so far is Alternating-time Temporal Logic (ATL) [1], in which one can specify what objectives coalitions of agents can achieve. Several extensions were introduced (ATL^*, game logics...), but all of these logics fail to model non-cooperative situations where agents follow individual objectives. It is well known that studying this kind of situation requires solution concepts from game theory, such as Nash equilibria, that cannot be expressed in ATL or its extensions.

To address this shortcoming, Chatterjee, Henzinger and Piterman recently introduced Strategy Logic (SL) [9]. This logic subsumes all extensions of ATL, and because it considers strategies as first-order citizens in the language, it can express fundamental game-theoretic concepts such as Nash Equilibria or dominated strategies. SL has recently been extended and intensively studied [19, 17, 18]. Relevant fragments enjoying nice computational characteristics have been identified. In particular, the syntactic fragment SL[1G] (One-Goal Strategy Logic) is strictly more expressive than ATL^*, but not computationally more expensive [17].

However, despite its great expressiveness, there is one fundamental feature of most real-life situations that SL lacks, which is imperfect information. An agent has imperfect information if she does not know the exact state of the system at every moment, but only has access to an approximation of it. Considering agents with imperfect information raises two major theoretical issues. The first one concerns strategizing under imperfect information. Indeed, in this context an agent's strategy must prescribe the same choice in all situations that are indistinguishable to the agent. Such strategies are called *uniform strategies*, and this requirement deeply impacts the task of computing strategies [21]. The second main theoretical challenge relates to uncertainty, deeply intertwined with imperfect information, and it consists of representing and reasoning about agents' knowledge. Over the past decades, much effort has been put into devising logical systems that address this issue, first in static settings [10] and later adding dynamics [11, 5].

J. Gutierrez, F. Mogavero, A. Murano, and M. Wooldridge (Eds.):
3rd International Workshop on Strategic Reasoning 2015 (SR15)

Concerning ATL, many variants have been introduced that deal with imperfect information [12, 14, 22, 13]. Some of these numerous logics deal with strategizing under imperfect information, some with reasoning about knowledge; because it is not natural to reason about the knowledge of agents with imperfect information without treating the strategic aspects accordingly, as argued in [14], some treat both aspects. But there still remain a number of logics that do so, and that essentially differ in the semantics of the strategic operator: how much memory do agents have? should agents simply have a strategy to achieve some goal? Or should they *know that there is* a strategy to do so? Or *know a strategy* that works? The two last notions are usually referred to as *de dicto* and *de re* strategies, respectively [14].

About SL, very few works have considered imperfect information. [2] and [8] propose epistemic extensions of SL, but they do not require strategies to be uniform *i.e.* being consistent with the agents' information. In [3], an epistemic strategy logic is proposed in which uniform strategies are considered, and interestingly the *de re* semantics of the strategic operator can be expressed in the *de dicto* semantics, providing some flexibility. However, how much memory strategies use , and whether they should be uniform or not, still has to be hardwired in the semantics.

In this work, we propose yet another epistemic strategy logic, with the purpose of getting rid of the constraint of enforcing what kind of strategies are to be used at the semantic level. To do so, we first develop a "branching-time" version of SL with perfect information. In SL, temporal operators are allowed only when all strategies are fixed, leaving only one possible play. We relax this constraint by introducing a *path quantifier*, which quantifies over the different outcomes defined by a partial assignment of strategies to the agents. This enables the comparison of the various outcomes of a strategy. Because it will be important, for instance to express the uniformity of a strategy, to consider all the possible outcomes of a strategy assigned to an agent a, we need a way to remove in an assignment the bindings of all agents but a. We thus introduce an *unbinding operator*. We call the resulting logic Branching-time Strategy Logic (BSL), and we prove by providing linear translations in both directions that it has the same expressive power and same computational complexity as SL. We also present a variant of BSL, called BSL^+, which can in addition refer to the actions chosen by each agent at each moment, and we conjecture that it is strictly more expressive than SL and BSL. Next, we define our Epistemic Strategy Logic (ESL) by further extending BSL with distributed knowledge operators, the semantics of which rely on equivalence relations on partial plays. We do not change the semantics of the strategy quantifier to require them to be uniform, or *de re*, or *de dicto*, or *memoryless*, but we rather show that all of these properties of strategies can be expressed in the language, which thus subsumes most, if not all, the variants of epistemic strategic logics with imperfect information that we know about.

The paper is structured as follows. In Section 2 we recall the models, syntax and semantics of SL. In Section 3, we define BSL and BSL^+, and we prove that SL and BSL are equiexpressive. We then present ESL in Section 4, where we also show how it can express various classic properties of strategies. We conclude and discuss future work in Section 5. Some proofs are omitted by lack of space.

2 Preliminaries

Let AP be a countable non-empty set of *atomic propositions*, Ag a non-empty finite set of *agents* and Ac a non-empty finite set of *actions*. We let $Dc = Ac^{Ag}$ be the set of possible *decisions*. For $d \in Dc$ and $a \in Ag$, $d(a)$ is the action taken by Agent a in decision d.

2.1 Concurrent game structures

A *concurrent game structure* (CGS) is a tuple $G = (Q, \delta, s_\iota, \mu)$, where Q is a countable non-empty set of *states*, $\delta : Q \times Dc \to Q$ is a *transition function*, s_ι is the *initial state* and $\mu : Q \to 2^{AP}$ is a *valuation function*. A *path* is an infinite word $\pi = s_0(d_1, s_1)\ldots \in Q \cdot (Dc \times Q)^\omega$ such that for all $i \geq 0$, $s_{i+1} = \delta(s_i, d_{i+1})$, and an *initial path* ρ is a finite prefix of a path. In the following, we shall write $s_0 d_1 s_1 \ldots$ instead of $s_0(d_1, s_1)\ldots$, and similarly for initial paths. For a state s we denote by $Paths_\omega(s)$ (resp. $Paths_*(s)$) the set of paths (resp. initial paths) that start in s, *i.e.* for which $s_0 = s$. We also let $Paths_\omega$ (resp. $Paths_*$) be the set of all paths (resp. initial paths). For a path $\pi = s_0 d_1 s_1 \ldots$, for $i, j \geq 0$, we let $\pi[i] := s_i$, $\pi_{\leq i} := s_0 \ldots d_i s_i$, $\pi_{\geq i} := s_i d_{i+1} s_{i+1} \ldots$ and $\pi[i, j] := s_i d_{i+1} \ldots d_j s_j$. For an initial path $\rho = s_0 \ldots d_n s_n$, $\text{last}(\rho) := s_n$ is its last state and $|\rho| := n$ is the index of its last state. Given two initial paths $\rho = s_0 d_1 s_1 \ldots d_n s_n$ and $\rho' = s'_0 d'_1 s'_1 \ldots d'_m s'_m$ such that $s_n = s'_0$, we let $\rho \cdot \rho' := s_0 d_1 s_1 \ldots d_n s_n d'_1 s'_1 \ldots d'_m s'_m$ be their concatenation.

A *strategy* is a total function $\sigma : Paths_* \to Ac$ that assigns an action to each initial path, and we let *Str* be the set of all strategies. Also, given a strategy σ and an initial path $\rho \in Paths_*$ ending in state s, we define the ρ*-translation* of σ as the strategy σ^ρ such that for all initial paths $\rho' \in Paths_*(s)$, $\sigma^\rho(\rho') := \sigma(\rho \cdot \rho')$, and for all initial paths $\rho' \in Paths_*(s')$ with $s' \neq s$, $\sigma^\rho(\rho') = \sigma(\rho')$.

Let *Var* be a countably infinite set of *variables*. An *assignment* is a partial function $\chi : Ag \cup Var \rightharpoonup Str$, assigning to each agent and variable in its domain a strategy. For an assignment χ, an agent a and a strategy σ, $\chi[a \mapsto \sigma]$ is the assignment of domain $dom(\chi) \cup \{a\}$ that maps a to σ and is equal to χ on the rest of its domain, and similarly for $\chi[x \mapsto \sigma]$ where x is a variable; also, $\chi[a \mapsto ?]$ is the assignment of domain $dom(\chi) \setminus \{a\}$, on which it is equal to χ. Given an assignment χ and a state s, we define the *outcome* of χ in s, written $\text{Out}(s, \chi)$, as the set of paths $\pi = s_0 d_1 s_1 \ldots$ such that $s_0 = s$, and for all $k \geq 0$, for every agent a in the domain of χ, $d_{k+1}(a) = \chi(a)(\pi_{\leq k})$. We say that an assignment χ is *complete* if it assigns a strategy to each agent, *i.e.* $Ag \subseteq dom(\chi)$. Given an assignment χ and an initial path ρ ending in state s, we define the ρ-translation of χ as the assignment χ^ρ such that $dom(\chi^\rho) = dom(\chi)$, and for all $l \in dom(\chi^\rho)$, $\chi^\rho(l) := \chi(l)^\rho$ (l being either a variable or an agent).

Finally, we want (some of) our logics to be able to talk about the precise actions taken by agents. To do so, we consider the following set of *action propositions*: $AcP := \{p^a_c \mid c \in Ac \text{ and } a \in Ag\}$, and we let $AP^+ := AP \uplus AcP$. In the following, we will therefore always assume that CGSs are *unfolded*, such that each state s is reached by one unique transition through some decision d_s, except the initial state s_ι which has no incoming transition. We can thus extend the valuation function μ into μ^+ as follows: $\mu^+(s_\iota) := \mu(s_\iota)$, and for every state $s \neq s_\iota$, $\mu^+(s) := \mu(s) \cup \{p^a_{d_s(a)} \mid a \in Ag\}$.

2.2 Strategy Logic

We recall the syntax and semantics of Strategy Logic (SL). First, the set of formulas in SL is given by the following grammar:

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

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

Notice that SL-formulas cannot talk about agents' actions.

We define $\top$ as $p \vee \neg p$. Dual operators can be defined as usual: $\bot := \neg\top, \varphi \wedge \varphi' := \neg(\neg\varphi \vee \neg\varphi'), \varphi\mathbf{R}\varphi' := \neg(\neg\varphi\mathbf{U}\neg\varphi')$ and $[\![x]\!]\varphi := \neg\langle\langle x\rangle\rangle\neg\varphi$, and we also define the classic temporal operators "eventually" and "always": $\mathbf{F}\varphi := \top\mathbf{U}\varphi$, and $\mathbf{G}\varphi := \varphi\mathbf{U}\bot$. Recall that $\langle\langle x\rangle\rangle$ is the *strategy quantifier*, and (a,x) is the *binding operator*: $\langle\langle x\rangle\rangle\varphi$ reads as "there exists a strategy x such that φ", and $(a,x)\varphi$ reads as "φ holds after agent a is bound to the strategy denoted by x".

For a formula φ, $\mathit{Free}(\varphi) \subseteq \mathit{Var}$ is the set of *free variables* in φ, *i.e.* the set of variables x that occur in φ without being under the scope of some quantification $\langle\langle x\rangle\rangle$. In the following, given a formula φ, an *assignment for* φ refers to an assignment χ such that $\mathit{Free}(\varphi) \subseteq \mathit{dom}(\chi)$.

Let φ be an SL-formula. Given a CGS $G = (Q,\delta,s_\iota,\mu)$, an assignment χ for φ and a state $s \in Q$, the semantics of φ in G with assignment χ at state s is defined inductively as follows:

$G,\chi,s \models_{\mathsf{SL}} p$ if $p \in \mu(s)$
$G,\chi,s \models_{\mathsf{SL}} \neg\varphi$ if $G,\chi,s \not\models_{\mathsf{SL}} \varphi$
$G,\chi,s \models_{\mathsf{SL}} \varphi \vee \varphi'$ if $G,\chi,s \models_{\mathsf{SL}} \varphi$ or $G,\chi,s \models_{\mathsf{SL}} \varphi'$
$G,\chi,s \models_{\mathsf{SL}} \langle\langle x\rangle\rangle\varphi$ if there exists $\sigma \in \mathit{Str}(s)$ such that $G,\chi[x \mapsto \sigma],s \models_{\mathsf{SL}} \varphi$
$G,\chi,s \models_{\mathsf{SL}} (a,x)\varphi$ if $G,\chi[a \mapsto \chi(x)],s \models_{\mathsf{SL}} \varphi$

If, in addition, χ is complete, then

$G,\chi,s \models_{\mathsf{SL}} \mathbf{X}\varphi$ if $G,\chi^{\pi_{\leq 1}},\pi[1] \models_{\mathsf{SL}} \varphi$, where π is the only path in $\mathrm{Out}(s,\chi)$
$G,\chi,s \models_{\mathsf{SL}} \varphi\mathbf{U}\varphi'$ if there is $i \geq 0$ such that, letting π be the only path in $\mathrm{Out}(s,\chi)$, $G,\chi^{\pi_{\leq i}},\pi[i] \models_{\mathsf{SL}} \varphi'$, and for all $0 \leq j < i$, $G,\chi^{\pi_{\leq j}},\pi[j] \models_{\mathsf{SL}} \varphi$.

Finally, we define an *SL-sentence* to be an SL-formula φ such that $\mathit{Free}(\varphi) = \emptyset$ and every temporal operator in φ is under the scope of a binding for each agent.

3 Branching-time Strategy Logic

We now present a first extension of Strategy Logic. In SL, temporal operators are allowed only when every agent has been assigned a strategy, which leaves only one possible outcome. Here we relax this constraint: a temporal formula can be evaluated on the outcome of a *partial* strategy assignment. The outcome of such an assignment is a tree that contains all paths corresponding to all possible completions of the assignment, which is why we use the path quantification of branching-time temporal logic. We also add the *unbinding* operator as considered in *e.g.* [16], making it possible to unbind an agent from its strategy. We first show that the logic thus obtained, called BSL, has the same expressivity as SL, by providing linear translations in both directions. The unbinding operator is thus just convenient syntactic sugar. Then we further extend BSL by allowing it to refer to actions taken by agents, and obtain the logic BSL^+ that, we postulate, is strictly more expressive than SL and BSL. BSL has two advantages: first, the semantics is slightly cleaner than that of SL, as it is defined for all formulas and all assignments; second, the unbinding operator makes it possible to easily express that we unbind an agent, at no complexity cost. Finally, because it can explicitly refer to actions and consider outcomes of partial assignments, it is possible in BSL^+ to express properties of strategies, such as being memoryless or uniform, as we show in Section 4.

3.1 Syntax

The syntax of BSL adds two operators to SL. First, the *path quantifier*, borrowed from classic branching-time temporal logics: $\mathbf{E}\psi$ intuitively reads as "there exists an outcome of the currently fixed strategies in which φ holds". Second, the *unbinding operator*: $(a,?)\varphi$ means "φ holds after Agent a has been unbound from her strategy, if any". We define two variants, one (BSL) where formulas cannot talk about the actions taken by the agents, and one (BSL^+) where they can. Also, as for CTL^*, we find it convenient to distinguish between state and path formulas. Finally, the set of BSL-formulas (resp. BSL^+-formulas) is the set of state formulas given by the following grammar:

$$\begin{array}{ll} \text{State formulas:} & \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\langle x \rangle\rangle\varphi \mid (a,x)\varphi \mid (a,?)\varphi \mid \mathbf{E}\psi \\ \text{Path formulas:} & \psi ::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\psi \mid \psi\mathbf{U}\psi, \end{array}$$

where $p \in AP$ (resp. $p \in AP^+$), $x \in Var$ and $a \in Ag$.

Observe that $\mathsf{BSL} \subset \mathsf{BSL}^+$. In addition to the shorthand defined in Section 2.2, we also define the dual of the path quantifier: $\mathbf{A}\varphi := \neg\mathbf{E}\neg\varphi$. Finally, we write BSL^+_ψ (resp. BSL_ψ) for the set of BSL^+ (resp. BSL) path formulas.

3.2 Semantics

State formulas are evaluated in a state of (the unfolding of) a CGS, and path formulas in paths. Since BSL is a syntactical fragment of BSL^+, it is enough to define the latter's semantics.

Let $\varphi \in \mathsf{BSL}^+$ be a state formula (resp. let $\psi \in \mathsf{BSL}^+_\psi$ be a path formula), and let $G = (Q, \delta, q_\iota, \mu)$ be a CGS. Let $s \in G$ be a state, $\pi \in Paths_\omega$ a path, and let χ be an assignment for φ (resp. for ψ). The semantics of BSL^+ is defined inductively as follows:

$$\begin{array}{lll}
G,\chi,s \models_{\mathsf{BSL}} p & \text{if} & p \in \mu^+(s) \\
G,\chi,s \models_{\mathsf{BSL}} \neg\varphi & \text{if} & G,\chi,s \not\models_{\mathsf{BSL}} \varphi \\
G,\chi,s \models_{\mathsf{BSL}} \varphi \vee \varphi' & \text{if} & G,\chi,s \models_{\mathsf{BSL}} \varphi \text{ or } G,\chi,s \models_{\mathsf{BSL}} \varphi' \\
G,\chi,s \models_{\mathsf{BSL}} \langle\langle x \rangle\rangle\varphi & \text{if} & \text{there exists } \sigma \in Str \text{ such that } G,\chi[x \mapsto \sigma],s \models_{\mathsf{BSL}} \varphi \\
G,\chi,s \models_{\mathsf{BSL}} (a,x)\varphi & \text{if} & G,\chi[a \mapsto \chi(x)],s \models_{\mathsf{BSL}} \varphi \\
G,\chi,s \models_{\mathsf{BSL}} (a,?)\varphi & \text{if} & G,\chi[a \mapsto ?],s \models_{\mathsf{BSL}} \varphi \\
G,\chi,s \models_{\mathsf{BSL}} \mathbf{E}\psi & \text{if} & \text{there exists } \pi \in \mathrm{Out}(s,\chi) \text{ such that } G,\chi,\pi \models_{\mathsf{BSL}} \psi \\
G,\chi,\pi \models_{\mathsf{BSL}} \varphi & \text{if} & G,\chi,\pi[0] \models_{\mathsf{BSL}} \varphi \\
G,\chi,\pi \models_{\mathsf{BSL}} \neg\psi & \text{if} & G,\chi,\pi \not\models_{\mathsf{BSL}} \psi \\
G,\chi,\pi \models_{\mathsf{BSL}} \psi \vee \psi' & \text{if} & G,\chi,\pi \models_{\mathsf{BSL}} \psi \text{ or } G,\chi,\pi \models_{\mathsf{BSL}} \psi' \\
G,\chi,\pi \models_{\mathsf{BSL}} \mathbf{X}\psi & \text{if} & G,\chi^{\pi_{\leq 1}},\pi_{\geq 1} \models_{\mathsf{BSL}} \psi \\
G,\chi,\pi \models_{\mathsf{BSL}} \psi\mathbf{U}\psi' & \text{if} & \text{there is } i \geq 0 \text{ such that } G,\chi^{\pi_{\leq i}},\pi_{\geq i} \models_{\mathsf{BSL}} \psi' \text{, and} \\
& & \text{for all } 0 \leq j < i,\ G,\chi^{\pi_{\leq j}},\pi_{\geq j} \models_{\mathsf{BSL}} \psi
\end{array}$$

The semantics of the unbinding operator comes without surprise: $(a,?)\varphi$ holds in an assignment if φ holds after we have removed a from the domain of this assignment. For the path quantifier, $\mathbf{E}\psi$ holds if there is an outcome of the current assignment in the current state that verifies ψ.

For a BSL^+-formula φ, we write $G,\chi \models_{\mathsf{BSL}} \varphi$ if $Paths_\omega(s_\iota),\chi,s_\iota \models_{\mathsf{BSL}} \varphi$. Classically, a *$BSL^+$-sentence* is a BSL^+-formula without free variables, and similarly for BSL-sentences. For a BSL^+-sentence φ, we write $G \models_{\mathsf{BSL}} \varphi$ if $G,\chi \models_{\mathsf{BSL}} \varphi$ for any assignment χ.

3.3 Expressivity of BSL

We establish that BSL and SL have the same expressivity, and postulate that BSL^+ is strictly more expressive than both logics. First, given two logics $\mathscr{L}$ and $\mathscr{L}'$ whose sentences are evaluated on CGS's, we say that *$\mathscr{L}'$ subsumes $\mathscr{L}$*, written $\mathscr{L} \preccurlyeq \mathscr{L}'$, if for every $\mathscr{L}$-sentence φ there is an $\mathscr{L}'$-sentence φ' such that, for every CGS G, $G \models \varphi$ if, and only if, $G \models \varphi'$. We say that $\mathscr{L}$ and $\mathscr{L}'$ are *equiexpressive* if $\mathscr{L} \preccurlyeq \mathscr{L}'$ and $\mathscr{L}' \preccurlyeq \mathscr{L}$. We say that *$\mathscr{L}'$ strictly subsumes $\mathscr{L}$*, written $\mathscr{L} \prec \mathscr{L}'$, if $\mathscr{L} \preccurlyeq \mathscr{L}'$ and $\mathscr{L}' \not\preccurlyeq \mathscr{L}$.

We start with the easy direction, showing that BSL subsumes SL.

Definition 1 *The translation* $\mathrm{tr} : SL \to BSL$ *is defined by induction as follows:*

$$\begin{array}{lllllllll} \mathrm{tr}(p) & = & p & \mathrm{tr}(\neg\varphi) & = & \neg\mathrm{tr}(\varphi) & \mathrm{tr}(\varphi\vee\varphi') & = & \mathrm{tr}(\varphi)\vee\mathrm{tr}(\varphi') \\ \mathrm{tr}(\mathbf{X}\varphi) & = & \mathbf{EX}\mathrm{tr}(\varphi) & & & & \mathrm{tr}(\varphi\mathbf{U}\varphi') & = & \mathbf{E}\mathrm{tr}(\varphi)\mathbf{U}\mathrm{tr}(\varphi') \\ \mathrm{tr}(\langle\langle x\rangle\rangle\varphi) & = & \langle\langle x\rangle\rangle\mathrm{tr}(\varphi) & & & & \mathrm{tr}((a,x)\varphi) & = & (a,x)\mathrm{tr}(\varphi) \end{array}$$

The following proposition easily follows from the fact that a complete assignment defines a unique path from any state.

Proposition 1 *For every CGS G, for every formula* $\varphi \in SL$*, assignment* χ *for* φ *state* $s \in G$ *such that* $G,\chi,s \models_{SL} \varphi$ *is defined, it holds that* $G,\chi,s \models_{SL} \varphi$ *if, and only if,* $G,\chi,s \models_{BSL} \mathrm{tr}(\varphi)$.

Proof sketch We only treat the case of the "next" operator, the one for "until" is similar and all the others are trivial. Assume that $G,\chi,s \models_{\mathsf{SL}} \mathbf{X}\varphi$ is defined. This means that χ is a complete assignment, hence $\mathrm{Out}(s,\chi)$ is a singleton, and the result follows from the semantics of SL and BSL. □

We now show that SL also subsumes BSL. Indeed, the path quantifier can be simulated by a series of existential strategy quantifications and corresponding bindings for the agents whose strategies are undefined in the current assignment. Concerning the unbinding operator, the idea is to remember, along the translation, which agents have been unbound, and use this information to correctly translate path quantifiers, as described above. Formally, we define a translation from BSL to SL, parameterized by the set of agents who are "currently" not bound to a strategy.

Definition 2 *Let* $A \subseteq Ag$*. The translations* $\mathrm{tr}'_A : BSL \to SL$ *and* $\mathrm{tr}'^{\psi}_A : BSL_\psi \to SL$ *are defined by mutual induction as follows:*

$$\begin{array}{llllll} \mathrm{tr}'_A(p) & = & p & \mathrm{tr}'^{\psi}_A(\psi) & = & \mathrm{tr}'_A(\varphi) \\ \mathrm{tr}'_A(\neg\varphi) & = & \neg\mathrm{tr}'_A(\varphi) & \mathrm{tr}'^{\psi}_A(\neg\psi) & = & \neg\mathrm{tr}'^{\psi}_A(\psi) \\ \mathrm{tr}'_A(\varphi\vee\varphi') & = & \mathrm{tr}'_A(\varphi)\vee\mathrm{tr}'_A(\varphi') & \mathrm{tr}'^{\psi}_A(\psi\vee\psi') & = & \mathrm{tr}'^{\psi}_A(\psi)\vee\mathrm{tr}'^{\psi}_A(\psi') \\ \mathrm{tr}'_A(\langle\langle x\rangle\rangle\varphi) & = & \langle\langle x\rangle\rangle\mathrm{tr}'_A(\varphi) & \mathrm{tr}'^{\psi}_A(\mathbf{X}\psi) & = & \mathbf{X}\mathrm{tr}'^{\psi}_A(\psi) \\ \mathrm{tr}'_A((a,x)\varphi) & = & (a,x)\mathrm{tr}'_{A\setminus\{a\}}(\varphi) & \mathrm{tr}'^{\psi}_A(\psi\mathbf{U}\psi') & = & \mathrm{tr}'^{\psi}_A(\psi)\mathbf{U}\mathrm{tr}'^{\psi}_A(\psi') \\ \mathrm{tr}'_A((a,?)\varphi) & = & \mathrm{tr}'_{A\cup\{a\}}(\varphi) & & & \\ \mathrm{tr}'_A(\mathbf{E}\psi) & = & \multicolumn{4}{l}{\langle\langle x_1\rangle\rangle\ldots\langle\langle x_k\rangle\rangle(a_{i_1},x_1)\ldots(a_{i_k},x_k)\mathrm{tr}'^{\psi}_A(\psi),} \\ & & \multicolumn{4}{l}{\textit{where } x_1,\ldots,x_k \textit{ are fresh variables and } \{a_{i_1},\ldots,a_{i_k}\} = A.} \end{array}$$

First, observe that if p is a BSL-formula, then it is in AP and not in AcP, so that p is indeed an SL formula. Before establishing the correctness of the translation, we need the following lemma. It essentially says that the evaluation of a formula $\mathrm{tr}'_A(\varphi)$ in an assignment χ is independent of how χ is defined on A: for an agent $a \in A$, whether χ is defined on a or not, and in the former case how it is defined, is of no consequence as the translation tr'_A remembers that a is not supposed to be bound to a strategy.

Lemma 1 *Let G be a CGS,* $s \in G$ *a state,* $\varphi \in BSL$ *a state formula and* χ *an s-total assignment for* φ*. For all* $A \subseteq Ag$*,* $\{a_{i_1},\ldots,a_{i_k}\} \subseteq A$ *and for all* $\sigma_1,\ldots,\sigma_k \in \mathrm{Str}(s)$*, letting* $\chi_1 = \chi[a_{i_1} \mapsto \sigma_1,\ldots,a_{i_k} \mapsto \sigma_k]$ *and* $\chi_2 = \chi[a_{i_1} \mapsto ?,\ldots,a_{i_k} \mapsto ?]$*, it holds that:*

P1: $G,\chi,s \models_{SL} \mathrm{tr}'_A(\varphi)$ *if, and only if,* $G,\chi_1,s \models_{SL} \mathrm{tr}'_A(\varphi)$*, and*

P2: $G,\chi,s \models_{SL} \mathrm{tr}'_A(\varphi)$ *if, and only if,* $G,\chi_2,s \models_{SL} \mathrm{tr}'_A(\varphi)$.

Proposition 2 *Let G be a CGS. For every state formula* $\varphi \in BSL$*, assignment* χ *for* φ *and state* $s \in G$*, it holds that* $G,\chi,s \models_{BSL} \varphi$ *if, and only if,* $G,\chi,s \models_{SL} \mathrm{tr}'_{Ag\setminus dom(\chi)}(\varphi)$.

We can now prove that SL and BSL have the same expressivity on the level of sentences.

Theorem 1 *SL and BSL are equiexpressive, with linear translations in both directions.*

Proof We first prove that SL $\preccurlyeq$ BSL. Let φ be an SL-sentence. Clearly, $\mathrm{tr}(\varphi)$ is a BSL-sentence. Let G be a CGS with initial state s_ι, and let χ be any assignment. By definition, $G \models_{\mathsf{SL}} \varphi$ iff $G, \chi, s_\iota \models_{\mathsf{SL}} \varphi$. By Proposition 1, $G, \chi, s_\iota \models_{\mathsf{SL}} \varphi$ iff $G, \chi, s_\iota \models_{\mathsf{BSL}} \mathrm{tr}(\varphi)$, and by definition, the latter is equivalent to $G \models_{\mathsf{BSL}} \mathrm{tr}(\varphi)$.

Now, to prove that BSL $\preccurlyeq$ SL, let φ be a BSL-sentence, and let $\varphi' = \mathrm{tr}'_{Ag}(\varphi)$. Observe that φ' is an SL-sentence: indeed, every temporal operator in φ is under the scope of some path quantifier, and by definition of tr'_{Ag}, every temporal operator in φ' is thus under the scope of a binding for each agent. Now, let G be a CGS and χ an assignment such that $Ag \setminus dom(\chi) = Ag$. By definition, $G \models_{\mathsf{BSL}} \varphi$ iff $G, \chi, s_\iota \models_{\mathsf{BSL}} \varphi$ (recall that since φ is a sentence, the choice of χ does not matter for the evaluation of φ). By Proposition 2, the latter is equivalent to $G, \chi, s_\iota \models_{\mathsf{SL}} \varphi'$, which by definition is equivalent to $G \models_{\mathsf{SL}} \varphi'$.

Concerning the size of the translations, the one of Definition 1 is clearly linear, and the one in Definition 2 is in $O(2|Ag||\varphi|)$, where $|Ag|$ is the number of agents and $|\varphi|$ the number of symbols in φ. The translation is thus linear in the size of the formula. $\square$

We can therefore transfer to BSL the following results known about SL [18]:

Corollary 1 *The model-checking problem for BSL is nonelementary decidable.*

Corollary 2 *The satisfiability problem for BSL is Σ^1_1-hard.*

On the other hand, because BSL$^+$ can express properties about the actions taken by agents, it should clearly be strictly more expressive than BSL and thus also SL, but we have not yet proved this.

Conjecture 1 *BSL$^+$ strictly subsumes BSL and SL.*

4 Epistemic Strategy Logic

In this section, we further extend the framework to account for imperfect information. For the logic to be expressive enough to express uniformity of strategies, we need to talk about actions played by the agents, and we therefore allow the use of atomic propositions in AcP.

4.1 Syntax

We add distributed knowledge operators to the language, one for each group of agents. The syntax of ESL is therefore described by the following grammar:

$$\text{State formulas:} \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{E}\varphi \mid \langle\langle x\rangle\rangle\varphi \mid (a,x)\varphi \mid (a,?)\varphi \mid D_A\varphi$$

$$\text{Path formulas:} \quad \psi ::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\psi \mid \psi\mathbf{U}\psi,$$

where $p \in AP^+$, $x \in Var$ and $A \subseteq Ag$.

We define, for each $a \in Ag$, $K_a\varphi := D_{\{a\}}\varphi$, and as for BSL and BSL$^+$, we write ESL$_\psi$ for the set of ESL-path formulas.

4.2 Semantics

To represent the agents' imperfect information about the current situation in the game, we add binary indistinguishability relations in CGSs. Most works consider equivalence relations on states, which are extended to initial paths according to how much memory agents are supposed to have. Because in this work we do not want to make any such assumptions, we adopt a more general approach and directly take equivalence relations on initial paths.

We call *imperfect information concurrent game structure* (ICGS) a tuple $G_i = (G, \{\sim_a\}_{a\in Ag})$, where G is a CGS and for each $a \in Ag$, $\sim_a \subseteq (2^{AP^+})^* \times (2^{AP^+})^*$ is an indistinguishability equivalence relation for Agent a. For $A \subseteq Ag$, we let $\sim_A := \cap_{a\in A} \sim_a$: it is the *distributed knowledge relation* of agents in A. Given two initial paths $\rho = s_0 d_1 s_1 \dots d_n s_n$ and $\rho' = s'_0 d_1 s'_1 \dots d_m s'_m$ and a set of agents $A \subseteq Ag$, we shall write $\rho \sim_A \rho'$ whenever $\mu^+(s_0)\dots\mu^+(s_n) \sim_A \mu^+(s'_0)\dots\mu^+(s'_m)$, *i.e.* when the sequences of extended valuations along the plays are related by $\sim_A$. As usual in epistemic logic, the intended meaning of $\rho \sim_a \rho'$ is that in initial path ρ, Agent a considers it possible that ρ' is the actual initial path.

Because agents may infer knowledge from what they recall of the past of an initial path, we cannot evaluate state formulas merely in states of the game as we do for BSL^+, but we evaluate them in initial paths instead. Also, in order not to forget the past when we consider outcomes of an assignment, we define for every initial path ρ and assignment χ, $\mathrm{Out}(\rho,\chi) := \{\rho\cdot\rho' \mid \rho' \in \mathrm{Out}(\mathrm{last}(\rho),\chi)\}$.

Let $\varphi \in \mathsf{ESL}$ be a state formula (resp. let $\psi \in \mathsf{ESL}_\psi$ be a path formula), and let $G = (Q, \delta, q_\iota, \mu)$ be a CGS. Let χ be an assignment for φ (resp. for ψ), let $\rho \in Paths_*$ be an initial path, $\pi \in Paths_\omega$ a path, and $i \geq 0$. The semantics of ESL is defined inductively as follows:

$G_i,\chi,\rho \models_{\mathsf{ESL}} p$	if	$p \in \mu^+(\mathrm{last}(\rho))$
$G_i,\chi,\rho \models_{\mathsf{ESL}} \neg\varphi$	if	$G_i,\chi,\rho \not\models_{\mathsf{ESL}} \varphi$
$G_i,\chi,\rho \models_{\mathsf{ESL}} \varphi \vee \varphi'$	if	$G_i,\chi,\rho \models_{\mathsf{ESL}} \varphi$ or $G_i,\chi,\rho \models_{\mathsf{ESL}} \varphi'$
$G_i,\chi,\rho \models_{\mathsf{ESL}} \langle\langle x\rangle\rangle\varphi$	if	there exists $\sigma \in Str$ such that $G_i,\chi[x \mapsto \sigma],\rho \models_{\mathsf{ESL}} \varphi$
$G_i,\chi,\rho \models_{\mathsf{ESL}} (a,x)\varphi$	if	$G_i,\chi[a \mapsto \chi(x)],\rho \models_{\mathsf{ESL}} \varphi$
$G_i,\chi,\rho \models_{\mathsf{ESL}} (a,?)\varphi$	if	$G_i,\chi[a \mapsto ?],\rho \models_{\mathsf{ESL}} \varphi$
$G_i,\chi,\rho \models_{\mathsf{ESL}} \mathbf{E}\psi$	if	there exists $\pi \in \mathrm{Out}(\rho,\chi)$ such that $G_i,\chi,\pi,\lvert\rho\rvert \models_{\mathsf{ESL}} \psi$
$G_i,\chi,\rho \models_{\mathsf{ESL}} D_A\varphi$	if	for every initial path $\rho' \in Paths_*$ such that $\rho \sim_A \rho'$, $G_i,\chi,\rho' \models_{\mathsf{ESL}} \varphi$
$G_i,\chi,\pi,i \models_{\mathsf{ESL}} \varphi$	if	$G_i,\chi,\pi_{\leq i} \models_{\mathsf{ESL}} \varphi$
$G_i,\chi,\pi,i \models_{\mathsf{ESL}} \neg\psi$	if	$G_i,\chi,\pi,i \not\models_{\mathsf{ESL}} \psi$
$G_i,\chi,\pi,i \models_{\mathsf{ESL}} \psi \vee \psi'$	if	$G_i,\chi,\pi,i \models_{\mathsf{ESL}} \psi$ or $G_i,\chi,\pi,i \models_{\mathsf{ESL}} \psi'$
$G_i,\chi,\pi,i \models_{\mathsf{ESL}} \mathbf{X}\psi$	if	$G_i,\chi^{\pi[i,i+1]},\pi,i+1 \models_{\mathsf{ESL}} \psi$
$G_i,\chi,\pi,i \models_{\mathsf{ESL}} \psi\mathbf{U}\psi'$	if	there is $j \geq i$ such that $G_i,\chi^{\pi[i,j]},\pi,j \models_{\mathsf{ESL}} \psi'$, and for all $i \leq k < j$, $G_i,\chi^{\pi[i,k]},\pi,k \models_{\mathsf{ESL}} \psi$

We now give an example of a property that can be expressed in ESL but not in SL, BSL or BSL^+. The property we consider is the uniformity property of strategies, which is central in the paradigm of imperfect information.

4.3 Properties of strategies

A *uniform strategy*, in the context of games with imperfect information, usually means a strategy that respects the player's information, *i.e.* a strategy that assigns the same action in situations that are indistinguishable to the player [4, 14]. In SL, temporal formulas being only evaluated in complete assignments, it is clear that one cannot compare several outcomes of a given strategy for a player, so that it is hopeless to express such uniformity properties. In BSL, one can consider all the possible outcomes of a strategy, but one cannot talk about the actions taken by agents, so that expressing that a strategy assigns the same action in different situations is not possible either. In BSL^+, we can refer to the precise actions taken by the agents, but we have no way of relating situations that are indistinguishable to an agent. However, as we show below, ESL is expressive enough for this sort of properties.

We define a notion of uniformity, that we call *weak uniformity*, and that asks for a strategy to be uniform on all its outcomes from the current situation.

Definition 3 *Let $G_i = (G, \{\sim_a\}_{a\in Ag})$ be an ICGS, let $\rho \in Paths_*$ be an initial path and $a \in Ag$ an agent. A strategy σ is* weakly uniform for a in ρ *if, for all initial paths $\rho' \in Out(\rho, [a \mapsto \sigma])$ and $\rho'' \in Paths_*$ such that $\rho' \sim_a \rho''$, $\sigma(\rho') = \sigma(\rho'')$.*

Now let us define the following ESL-formula.

Definition 4 *For each $a \in Ag$, we define the formula*

$$a\text{-}\mathtt{wUniform\text{-}aux} := \mathbf{AG}(\bigvee_{c\in Ac} K_a \mathbf{AX} p_c^a).$$

To understand the meaning of this formula, first observe that if an assignment χ binds an agent a to a strategy σ, *i.e.* $\chi(a) = \sigma$, then for every initial path $\rho \in Paths_*$, there is an action $c \in Ac$ such that p_c^a holds in all continuations of ρ of the form $\rho' = \rho \cdot ds$ that follow χ: this action is $\sigma(\rho) = d(a)$, the action played by Agent a in initial path ρ according to σ. Therefore, $G_i, \chi, \rho \models \mathbf{AX} p^a_{\sigma(\rho)}$. It follows that, when evaluated in an initial path ρ and assignment $[a \mapsto \sigma]$, where σ is a strategy, formula $a\text{-}\mathtt{wUniform}$ says that at every point of every outcome in $\mathrm{Out}(\rho, \chi)$, there is an action that Agent a plays in *all $\sim_a$-related nodes*. Let us fix an ICGS $G_i = (G, \{\sim_a\}_{a\in Ag})$.

Proposition 3 *For every initial path $\rho \in Paths_*$ and agent $a \in Ag$, a strategy σ is weakly uniform for Agent a in ρ if, and only if, $G_i, [a \mapsto \sigma], \rho \models a\text{-}\mathtt{wUniform\text{-}aux}$.*

However, $a\text{-}\mathtt{wUniform}$ only has the intended meaning in an assignment that does not bind any other agent: indeed, otherwise we would only have that the strategy considered is uniform on the subset of its outcomes that follow the strategies assigned to the other agents. Consider now the following formula:

Definition 5 *For each $a \in Ag$, noting $\{a_1, \ldots, a_k\} = Ag \setminus \{a\}$, we define the formula*

$$a\text{-}\mathtt{wUniform} := (a_1, ?) \ldots (a_k, ?) a\text{-}\mathtt{wUniform\text{-}aux}.$$

The following proposition holds:

Proposition 4 *For every initial path $\rho \in Paths_*$, assignment χ and agent $a \in Ag$, a strategy σ is weakly uniform for Agent a in ρ if, and only if, $G_i, \chi[a \mapsto \sigma], \rho \models a\text{-}\mathtt{wUniform}$.*

We now illustrate how various semantics of ATL with imperfect information can be expressed in ESL. We take the example of the ATL formula $\langle\langle A\rangle\rangle \mathbf{F} p$, where $A \subseteq Ag$. Assume that $A = \{a_1, \ldots, a_k\}$ and $Ag \setminus A = \{a_{k+1}, \ldots, a_n\}$. We consider three semantics: the basic one from [15], in which strategies are just required to be uniform, the *de dicto* semantics, where in addition the players must know that there is a strategy to achieve their goal, but may ignore what that strategy is, and the *de re* semantics, in which there must exist a strategy that the players know it ensures their goal (see [14], Sec. 3.2). With the first semantics, $\langle\langle A\rangle\rangle \mathbf{F} p$ would be translated in ESL as:

$$\langle\langle x_1\rangle\rangle \ldots \langle\langle x_k\rangle\rangle (a_1, x_1) \ldots (a_k, x_k)(\bigwedge_{1\le i\le k} a_i\text{-}\mathtt{wUniform} \wedge \mathbf{AF} p).$$

For the *de dicto* semantics, one would write instead:

$$D_A\langle\langle x_1\rangle\rangle \ldots \langle\langle x_k\rangle\rangle (a_1,x_1)\ldots(a_k,x_k)\big(\bigwedge_{1\leq i\leq k} a_i\text{-}\mathtt{wUniform} \wedge \mathbf{AF}p\big),$$

while for the *de re* semantics, one would write:

$$\langle\langle x_1\rangle\rangle \ldots \langle\langle x_k\rangle\rangle D_A (a_1,x_1)\ldots(a_k,x_k)\big(\bigwedge_{1\leq i\leq k} a_i\text{-}\mathtt{wUniform} \wedge \mathbf{AF}p\big).$$

One may object that the notion of weak uniformity we consider is too weak compared to the usual one, which is that a strategy should be equal on all pairs of related initial paths. We argue that it is enough for a strategy to be uniform on all the initial paths it may be involved in while evaluating the formula.

For instance, in the example above, the objective is $\mathbf{AF}p$, so that it is enough to ensure that strategies for the agents are uniform on their outcome: if a satisfying set of strategies contains one σ_i that is not defined uniformly on some initial paths that are outside its outcome, this σ_i can easily be turned into a uniform strategy in the usual sense, it will still satisfy the formula.

Should we consider a more complex objective, in particular involving knowledge, weak uniformity may not be sufficient though. Consider the ESL formula $\langle\langle x\rangle\rangle(a,x)\mathbf{AG}K_a\mathbf{AF}p$, where $a \in Ag$, which means that Agent a wants a strategy such that she always knows that p will eventually be reached. This objective not only considers outcomes of the strategy from the current situation, but also outcomes from initial paths equivalent to the latter outcomes. In this case, we could strengthen the requirement on Agent a's strategy by repeating the weak-uniformity requirement after each knowledge operator. In the example:

$$\langle\langle x\rangle\rangle(a,x)(a\text{-}\mathtt{wUniform} \wedge \mathbf{AG}K_a(a\text{-}\mathtt{wUniform} \wedge \mathbf{AF}p)).$$

Finally, observe that if we introduced an artificial agent a_{mem} associated to the relation that relates two initial paths if they end up in the same state, then the formula $a_{\text{mem}}\text{-}\mathtt{wUniform}$ would characterize strategies that are memoryless on their outcomes from the current initial path, in the sense that their definition only depends on the last state of each initial path.

5 Conclusion

We have enriched SL with two operators, the path quantifier and the unbinding operator, which are convenient but do not add expressivity in the perfect information case; interestingly though, they do not increase complexity either. In the context of imperfect information however, these operators together with knowledge operators and the ability to talk about actions, allowed us to express properties of strategies which are usually fixed in the semantics of the logics, such as being uniform, *de re*, *de dicto*, memoryless… This feature makes our Epistemic Strategy Logic able to deal with a vast class of agents without having to change the semantics, and thus unifies many of the previous proposals in the area.

Of course this comes at a price, and the model-checking problem for this logic is certainly undecidable with perfect-recall relations and several agents. We believe that the next steps are, first, to see whether the syntactical fragments studied for SL with perfect information, such as One-Goal or Boolean-Goal Strategy Logic, can be transferred to BSL and then to ESL, and see whether they enjoy better complexity properties. The second natural move would be to look at structures which are known to work well with multiple agents with imperfect information: hierarchical knowledge [7, 20], recurring common knowledge of the state [6]…

References

[1] Rajeev Alur, Thomas A. Henzinger & Orna Kupferman (2002): *Alternating-time temporal logic*. J. ACM 49(5), pp. 672–713, doi:10.1145/585265.585270. Available at `http://doi.acm.org/10.1145/585265.585270`.

[2] Francesco Belardinelli (2014): *Reasoning about Knowledge and Strategies: Epistemic Strategy Logic*. In: SR, pp. 27–33. Available at `http://dx.doi.org/10.4204/EPTCS.146.4`.

[3] Francesco Belardinelli (2015): *A Logic of Knowledge and Strategies with Imperfect Information*. In: Private communication.

[4] Johan van Benthem (2001): *Games in Dynamic-Epistemic Logic*. Bulletin of Economic Research 53(4), pp. 219–248, doi:10.1111/1467-8586.00133.

[5] Johan van Benthem (2011): *Logical dynamics of information and interaction*. Cambridge University Press.

[6] Dietmar Berwanger & Anup Basil Mathew (2014): *Games with recurring certainty*. In: Proceedings of SR 2014, pp. 91–96, doi:10.4204/EPTCS.146.12.

[7] R. Bozianu, C. Dima & C. Enea (2013): *Model Checking an Epistemic mu-calculus with Synchronous and Perfect Recall Semantics*. In: TARK'2013. Available at `http://arxiv.org/abs/1310.6434`.

[8] Petr Cermák, Alessio Lomuscio, Fabio Mogavero & Aniello Murano (2014): *MCMAS-SLK: A Model Checker for the Verification of Strategy Logic Specifications*. In: CAV, pp. 525–532. Available at `http://dx.doi.org/10.1007/978-3-319-08867-9_34`.

[9] Krishnendu Chatterjee, Thomas A. Henzinger & Nir Piterman (2010): *Strategy logic*. Inf. Comput. 208(6), pp. 677–693, doi:10.1016/j.ic.2009.07.004. Available at `http://dx.doi.org/10.1016/j.ic.2009.07.004`.

[10] Ronald Fagin, Joseph Y. Halpern, Yoram Moses & Moshe Y. Vardi (1995): *Reasoning about knowledge*. 4, MIT press Cambridge.

[11] Joseph Y. Halpern, Ron van der Meyden & Moshe Y. Vardi (2004): *Complete Axiomatizations for Reasoning about Knowledge and Time*. SIAM J. Comput. 33(3), pp. 674–703. Available at `http://dx.doi.org/10.1137/S0097539797320906`.

[12] W. van der Hoek & M. Wooldridge (2003): *Cooperation, knowledge, and time: Alternating-time Temporal Epistemic Logic and its applications*. Studia Logica 75(1), pp. 125–157, doi:10.1023/A:1026185103185.

[13] W. Jamroga & T. Ågotnes (2006): *What agents can achieve under incomplete information*. In: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, ACM, pp. 232–234.

[14] Wojciech Jamroga & Wiebe van der Hoek (2004): *Agents that Know How to Play*. Fundam. Inform. 63(2-3), pp. 185–219. Available at `http://iospress.metapress.com/content/xh738axb47d8rchf/`.

[15] Wojtek Jamroga (2003): *Some remarks on alternating temporal epistemic logic*. Proceedings of Formal Approaches to Multi-Agent Systems (FAMAS 2003), pp. 133–140.

[16] François Laroussinie & Nicolas Markey (2013): *Satisfiability of ATL with strategy contexts*. In: Proceedings Fourth International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2013, Borca di Cadore, Dolomites, Italy, 29-31th August 2013., pp. 208–223, doi:10.4204/EPTCS.119.18. Available at `http://dx.doi.org/10.4204/EPTCS.119.18`.

[17] Fabio Mogavero, Aniello Murano, Giuseppe Perelli & Moshe Y. Vardi (2012): *What Makes Atl* Decidable? A Decidable Fragment of Strategy Logic*. In: CONCUR 2012 - Concurrency Theory - 23rd International Conference, CONCUR 2012, Newcastle upon Tyne, UK, September 4-7, 2012. Proceedings, pp. 193–208. Available at `http://dx.doi.org/10.1007/978-3-642-32940-1_15`.

[18] Fabio Mogavero, Aniello Murano, Giuseppe Perelli & Moshe Y. Vardi (2014): *Reasoning About Strategies: On the Model-Checking Problem*. ACM Trans. Comput. Log. 15(4), pp. 34:1–34:47, doi:10.1145/2631917. Available at `http://doi.acm.org/10.1145/2631917`.

[19] Fabio Mogavero, Aniello Murano & Moshe Y. Vardi (2010): *Reasoning About Strategies*. In: *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010, December 15-18, 2010, Chennai, India*, pp. 133–144, doi:10.4230/LIPIcs.FSTTCS.2010.133. Available at http://dx.doi.org/10.4230/LIPIcs.FSTTCS.2010.133.

[20] A. Pnueli & R. Rosner (1989): *On the Synthesis of a Reactive Module*. In: *POPL'89*, pp. 179–190, doi:10.1145/75277.75293. Available at http://doi.acm.org/10.1145/75277.75293.

[21] John H. Reif (1984): *The complexity of two-player games of incomplete information*. *Journal of computer and system sciences* 29(2), pp. 274–301, doi:10.1016/0022-0000(84)90034-5.

[22] Pierre-Yves Schobbens (2004): *Alternating-time logic with imperfect recall*. *Electronic Notes in Theoretical Computer Science* 85(2), pp. 82–93.