Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Public and Private Affairs in Strategic Reasoning

Nathanaël Fijalkow[1], **Bastien Maubert**[2], **Aniello Murano**[2], **Sasha Rubin**[3] and **Moshe Vardi**[4]
[1]CNRS & LaBRI, France
[2]Università degli Studi di Napoli "Federico II", Italy
[3]University of Sydney, Australia
[4]Rice University, U.S.A.

Abstract

Do agents know each others' strategies? In multi-process software construction, each process has access to the processes already constructed; but in typical human-robot interactions, a human may not announce their strategy to the robot (indeed, the human may not even know their own strategy). This question has often been overlooked when modeling and reasoning about multi-agent systems. In this work we study how it impacts strategic reasoning.

To do so we consider Strategy Logic (SL), a well-established and highly expressive logic for strategic reasoning. Its usual semantics, which we call "white-box semantics", models systems in which agents "broadcast" their strategies. By adding imperfect information to the evaluation games for the usual semantics, we obtain a new semantics called "black-box semantics", in which agents keep their strategies private. We consider the model-checking problem and show that the black-box semantics has much lower complexity than white-box semantics for an important fragment of Strategy Logic.

1 Introduction

In multi-agent systems, strategies tell agents what actions to take in order to try and achieve their goals. The strategy that an agent decides or chooses to use at a point in time may or may not depend on the strategies chosen by the other agents. For example, in typical agent applications such as human-robot interaction, a human may not announce its strategy to the robot (indeed, the human may not even be aware of their own strategy); on the other hand, in multi-process software construction projects one might assume that each process knows the strategy of the other processes already constructed. The purpose of this paper is to encapsulate this distinction in Strategy Logic (SL), a powerful logical formalism able to express a wide range of game-theoretic strategic concepts, such as winning strategies and Nash equilibria (Mogavero et al. 2014).

There are a number of strategic logics of varying expressiveness and complexities, among which the influential ATL/ATL* (Alur, Henzinger, and Kupferman 2002), which extend the branching-time logics CTL/CTL* with strategic modalities, ATL with strategy contexts (Laroussinie and Markey 2015), which remembers past strategies when evaluating new strategic modalities, and Strategy Logic and its fragments (Chatterjee, Henzinger, and Piterman 2010; Mogavero et al. 2014; Mogavero et al. 2017; Belardinelli et al. 2019; Aminof et al. 2019; Cermák, Lomuscio, and Murano 2015), which treat strategies as first-order variables. These logics are typically interpreted over multi-agent transition systems (known as "concurrent game structures") in which agents take simultaneous actions resulting in the next state. The usual semantics of these logics is what we call "white-box", which means, intuitively, that agents announce their strategies. We illustrate this with a simple example.

Consider the game of matching pennies in which two players, Alice and Bob, simultaneously choose Heads or Tails, and the objective of Bob is to match Alice's choice. Consider the property "for every strategy of Alice there exists a strategy of Bob such that the pennies match", which we could write in SL as

$$\forall s_A \, \exists s_B \, (\text{Alice}, s_A)(\text{Bob}, s_B)\text{Match},$$

where (a, s) indicates that agent a is assigned strategy s. Under the usual interpretation of quantifiers, this property is true: for a strategy s_A of Alice, the same strategy for Bob provides a witness to the satisfaction of the formula. And indeed this formula holds true in the usual semantics of SL, which reflects the usual mathematical interpretation.

This interpretation, however, is not always the desired one. In the example, a strategy s_B for Bob exists for each strategy s_A of Alice, but s_B depends on Alice's strategy s_A, and Bob is not able to know which strategy to pick unless Alice first announces her strategy. In most agent-based scenarios, however, agents are assumed to choose their strategies in secret and not announce them to their opponents. In such settings, if one wants $\exists s_B$ to express not only the existence of a strategy, but the ability for the agent to choose it, then a different type of semantics for such strategic properties is required, in which Bob's strategy is not allowed to depend at all on Alice's strategy. We call this interpretation "black-box" semantics, in contrast to the usual "white-box" semantics where strategies are publicly announced.

In this work, we define Strategy Logic with black-box semantics, and study its expressiveness and complexity.

1.1 Related Work

One thread in the history of strategic logics for multi-agent systems is the search for an expressive logic with elementary complexity. ATL* can express the existence or absence of winning strategies for a coalition, and has elementary complexity: its model-checking problem is 2-EXPTIME-complete, i.e., no harder than solving games with LTL winning conditions (which can be expressed in ATL^*). Richer logics can naturally express game-theoretic concepts such as the existence of Nash equilibria, including ATL^* with strategy contexts and Strategy Logic. However, model checking these logics is nonelementary decidable, with one exponential per alternation between existential and universal strategy quantifiers (Laroussinie and Markey 2013; Mogavero et al. 2017).

Then started a line of work that aims at taming the complexity of Strategy Logic, both by studying fragments or considering variations on the semantics. One approach is to restrict to positional strategies, i.e., strategies that only look at the current state rather than current history, which leads to a PSPACE-complete model-checking problem (Čermák et al. 2018). However, positional strategies are not adequate for general LTL objectives – the memory required may be double-exponential in the size of the formula already for very simply formulas encoding the synthesis problem (Pnueli and Rosner 1989).

Concerning fragments, the One-Goal fragment of SL is computationally no harder than ATL^*, but it cannot express the existence of Nash equilibria. Other fragments have been introduced that can express it. One important fragment is Nested-Goals (Mogavero et al. 2014), whose formulas consist of a prefix of strategy quantifiers followed by a Boolean combination of *goals*, i.e., formulas of the form $\flat\psi$ where $\flat$ is a sequence of strategy assignments for all agents, and ψ is an LTL formula which can have nested goals as atomic propositions. New strategy quantifications inside goals are also allowed if they start independent subsentences. Nested-Goals SL under the white-box semantics is very expressive and can express most properties of interest (it subsumes ATL^*, it can express existence of Nash and subgame perfect equilibria), but its model-checking problem is nonelementary, as for the full logic (Mogavero et al. 2014).

One proposal to explain the source of complexity in such logics is that in the usual semantics of these logics, strategies quantified later on in the formula can depend on the entirety of strategies quantified before. Typically, in a formula of the form $\forall s\,\exists t\,\varphi$, the action chosen by strategy t in any history h can depend on the action chosen by s on h and its prefixes, but also *future* histories of the form hh', and even on *counterfactual* histories that are neither prefixes of h nor continuations thereof. The case was made that such dependencies are unnatural in many cases of interest, and that avoiding them may also reduce drastically the complexity. These dependencies were studied in depth in (Mogavero et al. 2014) and later in (Gardy, Bouyer, and Markey 2020). A semantics in which future and counterfactual dependencies can be forbidden was proposed for the Nested-Goals fragment, called behavioural semantics (Mogavero et al. 2017). This involves a heavy mathematical machinery called dependence maps, related to Skolem functions for first-order logic, and the complexity of which explains why it was only defined for the Nested-Goals fragment and not the full logic. We observe that although the behavioural semantics forbids future and counterfactual dependencies, it allows for concurrent ones, which is not natural in many settings (see the matching penny example for instance).

The concept of dependence between quantifiers has also been much studied in classical logic. Among the most influential formalisms one can cite branching quantifiers (Henkin and Karp 1965), independence-friendly logic (Hintikka 1998) and Dependence Logic (Väänänen 2007).

Our work can be seen as a novel approach to the problem of dependencies in Strategy Logic.

1.2 Contributions

We consider Branching-time Strategy Logic (BSL), which adds to SL a quantifier on possible outcomes of the strategies currently bound to agents. This quantifier can be simulated in classic SL with white-box semantics by quantifying on strategies for agents that are not bound to any strategy, but this no longer works with the black-box semantics of strategy quantification, hence our choice to consider BSL.

After recalling the usual Tarski-style white-box semantics of BSL, we propose an equivalent definition via evaluation games; see (Hintikka and Sandu 1997) for an overview of game-theoretic semantics. These are two-player games played between Eloise and Abelard, where Eloise tries to prove that the given formula is true in the given model, while Abelard tries to prove the opposite. A formula is defined to be true in a model if Eloise has a winning strategy in the corresponding evaluation game. Our new black-box semantics for BSL is then obtained by hiding the universally quantified strategies so that Eloise's strategies cannot depend on them when she chooses existentially quantified ones, thus obtaining an imperfect-information evaluation game, reminiscent of Dependence Logic's evaluation games with imperfect information (Väänänen 2007).

We then focus on the "Nested-Goals" fragment of BSL, which we call BSL[NG]. We show that BSL[NG] with black-box semantics is strictly more expressive than ATL^*, can express existence of Nash or subgame perfect equilibria, but still enjoys an elementary model-checking problem, which we prove to be 3-EXPTIME-complete. To obtain the upper-bounds we observe that under black-box semantics, existential and universal strategy quantifiers commute, and thus black-box BSL[NG] translates into white-box BSL[NG] with only one alternation in strategy quantifiers.

Finally we introduce and discuss an extension of BSL that combines the two semantics, white-box and black-box, in a single logic with two kinds of strategy quantifiers: public quantifiers from the white-box semantics, and private quantifiers from the black-box one.

Outline In Section 2 we recall the classic white-box semantics of BSL. In Section 3 we present an alternative, equivalent definition of this semantics via evaluation games, we define the new black-box semantics by adding imperfect information to those evaluation games, and we discuss this new semantics. We study the model-checking problem for BSL[NG] with black-box semantics in Section 4. In Section 5 we discuss a logic merging both semantics. We conclude in Section 6.

2 Classic Strategy Logic

2.1 Games

For the rest of the paper we fix a number of parameters: AP is a finite set of *atomic propositions* (denoted $p, q, r, \ldots$), Ag is a finite set of *agents* or *players* (denoted $a, b, c, \ldots$), Act is a finite set of *actions* (usually denoted α), and Var is a finite set of *strategy variables* (usually denoted s).

Definition 1 (Game). *A concurrent game structure $\mathcal{G}$ (or* game *for short) is a tuple (V, v_0, Δ, ℓ) where:*

- *V is the set of* positions*,*
- *$v_0 \in V$ is the* initial position*,*
- *$\Delta : V \times Act^{Ag} \to V$ is the* transition function*,*
- *$\ell : V \to 2^{AP}$ is the* labeling function*.*

Joint actions. In a position $v \in V$, each player $a \in \text{Ag}$ chooses an action $\alpha_a \in \text{Act}$, and the game proceeds to the position $\Delta(v, \boldsymbol{\alpha})$, where $\boldsymbol{\alpha} \in \text{Act}^{\text{Ag}}$ stands for the *joint action* $(\alpha_a)_{a\in\text{Ag}}$. Given a joint action $\boldsymbol{\alpha} = (\alpha_a)_{a\in\text{Ag}}$ and $a \in \text{Ag}$, we let $\boldsymbol{\alpha}_a$ denote α_a.

Plays and strategies. A *play* π is an infinite word over V, and a *history* h is a finite prefix of a play. For a play $\pi = v_1 v_2 \ldots$ and an index $i \in \mathbb{N}$, history $\pi_{\leq i} = v_1 \ldots v_i$ is the prefix of π up until position i. Plays is the set of plays, and Hist the set of histories. A *strategy* is a function $\sigma : \text{Hist} \to \text{Act}$, and Str is the set of all strategies. The length of a history h is written $|h|$ and its last position $\text{last}(h)$.

Assignments and bindings. An *assignment* is a partial function $\chi : \text{Var} \to \text{Str}$ that interprets strategy variables, and a *binding* is a partial function $\beta : \text{Ag} \to \text{Str}$ that binds some agents to strategies. We write $\chi[s \mapsto \sigma]$ for the assignment that maps s to σ and is equal to χ on the rest of its domain, and similarly for bindings. For a binding β we also let $\beta[a \mapsto \emptyset]$ be the binding that is undefined on a and otherwise equal to β. We say that an agent is *bound* by β if it is in its domain $\text{dom}(\beta)$.

Outcomes. Given a binding β and a history $h = h' \cdot v_0$ we let $\text{Out}(h, \beta)$ be the set of plays that can be obtained starting from h when each agent a bound by β plays according to $\beta(a)$, i.e., the set of plays of the form $h' \cdot v_0 v_1 v_2 \ldots$ where for each $i \geq 0$ there exists a joint action $\boldsymbol{\alpha}$ where $\boldsymbol{\alpha}_a = \beta(a)(h' \cdot v_0 \ldots v_{i-1})$ for each agent a bound by β, such that $v_{i+1} \in \Delta(v_i, \boldsymbol{\alpha})$.

Turn-based games. While concurrent game structures provide the models of Strategy Logic, we will also use *turn-based games* to define the semantics of the logics. These games, where players play in turns, can be seen as a particular case of the games defined above, but it is convenient to use specialized notations. In such cases the set of positions $V = \uplus_{a\in\text{Ag}} V_a$ is partitioned between the players, and when in position $v \in V_a$ player a directly chooses the next position. Transitions are thus described by a relation $\Delta \subseteq V \times V$. Histories and plays are as before, a strategy for a player a is a function $V^* V_a \to V$ that respects the transitions, and outcomes are plays that follows the fixed strategies.

2.2 Syntax of Strategy Logic

We first present the full Strategy Logic. We consider Branching-time Strategy Logic (BSL), the branching-time variant of SL (Knight and Maubert 2015; Fijalkow et al. 2018; Berthon et al. 2021), which contains an *outcome quantifier* that quantifies on possible outcomes given the current history and binding, as well as an *unbinding operator* that releases an agent from its current strategy. The main reason to consider BSL is that it is expressively equivalent to SL (Knight and Maubert 2015) but allows explicit quantification on outcomes, while in usual SL this has to be simulated by strategy quantifications and bindings. Such artificial strategy quantifications may increase the computational complexity of evaluating the formula. But using strategy quantifiers instead of outcomes quantifiers is also a problem when considering unorthodox semantics for strategy quantification as we do here.

Definition 2. *The syntax of BSL is given by the following grammar:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists s\, \varphi \mid (a, s)\varphi \mid (a, \emptyset)\varphi \mid \mathbf{E}\psi$$
$$\psi ::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\psi \mid \psi\mathbf{U}\psi$$

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

As usual we define $\top = p \vee \neg p$, $\varphi \wedge \varphi' = \neg(\neg\varphi \vee \neg\varphi')$, $\forall s\, \varphi = \neg\exists s\, \neg\varphi$, $\mathbf{A}\psi = \neg\mathbf{E}\neg\psi$, $\mathbf{F}\psi = \top\mathbf{U}\psi$ and $\mathbf{G}\psi = \neg\mathbf{F}\neg\psi$. Without loss of generality we will assume that each strategy variable s is quantified at most once in each formula, which allows us to talk about *existentially quantified* and *universally quantified* variables.

Formulas of type φ are called *history formulas*, those of type ψ are called *outcome formulas*. We define $\text{HistSub}(\varphi)$ and $\text{OutSub}(\varphi)$ as the sets of history and outcome subformulas of φ, respectively. We say that a strategy variable appears *free* in a formula if it appears (in a binding operator) out of the scope of a strategy quantifier. A *sentence* is a history formula without free variables.

Remark 1. *Note that when all agents are bound to a strategy, i.e., when β is defined for all agents, this determines a unique outcome: $Out(h, \beta)$ is a singleton for any history h. As a result $\mathbf{E}\psi$ and $\mathbf{A}\psi$ are equivalent when evaluated in such a complete binding (see Definition 4), and we may thus omit the outcome quantifier. For instance if we have $Ag = \{a, b\}$, we may write $\exists s \forall t\, (a, s)(b, t)\mathbf{X}p$ instead of $\exists s \forall t\, (a, s)(b, t)\mathbf{A}\mathbf{X}p$.*

We also introduce the Nested-Goals fragment of BSL. The idea of Nested-Goals is that strategy quantifiers come in blocks, each followed by a Boolean combination of *goals* (written $\flat$), which consist of a sequence of strategy bindings or unbindings, followed by an outcome quantifier and outcome formula, in such a way that each block of quantifiers starts a closed formula whose semantics is independent of previous quantifiers and bindings. Goals can also be nested, allowing the rebinding of agents to previously quantified strategies along an outcome formula. This very expressive fragment subsumes ATL^* and can in addition express complex game-theoretic notions such as existence of Nash Equilibria or Subgame Perfect Equilibria.

Definition 3. BSL[NG] *consists of formulas generated by the following grammar:*

$$\begin{aligned}\Phi &::= \exists s\, \Phi \mid \phi\\ \phi &::= \phi \vee \phi \mid \neg\phi \mid \flat\\ \flat &::= (a,s)\flat \mid (a,\emptyset)\flat \mid \mathbf{E}\psi\\ \psi &::= p \mid \flat \mid \Phi \mid \neg\psi \mid \psi\vee\psi \mid \mathbf{X}\psi \mid \psi\mathbf{U}\psi\end{aligned}$$

where $p \in AP$, $a \in Ag$ and $s \in Var$, with the following syntactic constraint: each subformula Φ of an outcome formula ψ must be a closed sentence, meaning that it either unbinds or rebinds to newly quantified strategies all agents that were bound above Φ, before each outcome quantifier $\mathbf{E}$. Such a formula is called an independent subsentence.

Since independent subsentences can be evaluated independently, we will sometimes treat them as atomic propositions. For instance when considering the model-checking problem, a bottom-up model-checking procedure can start by evaluating innermost independent subsentences and decorate the model with atomic propositions indicating where they hold. Seeing independent subsentences as atomic formulas leads to the *flat fragment* of BSL[NG], denoted by BSL[NG,f] and defined by the same grammar as above, except that Φ is removed from the production rule for outcome formulas, which becomes:

$$\psi ::= p \mid \flat \mid \neg\psi \mid \psi\vee\psi \mid \mathbf{X}\psi \mid \psi\mathbf{U}\psi$$

Example 1. *Write $Ag = \{a_1, \dots, a_n\}$, and consider that each agent a_i aims at making true some property expressed by an* LTL *formula ψ_i. Define the following formula*

$$\Phi_{NE} = \exists s_1 \dots \exists s_n\, \forall t \bigwedge_{i=1}^{n} (Ag_{-i}, \boldsymbol{s}_{-i})(a_i,t)\psi_i \to (Ag,\boldsymbol{s})\psi_i,$$

where $(Ag,\boldsymbol{s})$ is a shorthand for $(a_1,s_1)\dots(a_n,s_n)$ and, similarly, notation $(Ag_{-i},\boldsymbol{s}_{-i})$ is a shorthand for $(a_1,s_1)\dots(a_{i-1},s_{i-1})(a_{i+1},s_{i+1})\dots(a_n,s_n)$. Φ_{NE} expresses that there exists a Nash Equilibrium for objectives ψ_i. This formula is in BSL[NG], *and even in its flat fragment as it does not contain any strategy quantifier under an outcome quantifier (each ψ_i is a pure* LTL *formula).*

Consider now formula

$$\Phi_{SPE} = \exists s_1 \dots \exists s_n\, (Ag,\boldsymbol{s})\mathbf{AG}\, \forall t \bigwedge_{i=1}^{n} (a_i,t)\psi_i \to \psi_i,$$

which expresses that there exists a Subgame Perfect equilibrium. It is not in the Nested Goals fragment: indeed the subformula on the second line is of type Φ, but each conjunct only rebinds one agent (a_i) before the outcome quantifier[1]*, leaving other agents bound to strategies from the previous block of strategy quantifiers.*

One could consider rewriting into the equivalent formula

$$\Phi'_{SPE} = \exists s_1 \dots \exists s_n\, (Ag,\boldsymbol{s})\mathbf{AG}\, \forall t \bigwedge_{i=1}^{n} (Ag_{-i}, \boldsymbol{s}_{-i})(a_i,t)\psi_i \to (Ag,\boldsymbol{s})\psi_i,$$

but the latter is still not part of the Nested Goals fragment. Indeed here in the subformula of the second line each agent is bound again, but only one is bound to a newly quantified strategy. All the remaining agents are bound to strategies quantified in the first part of the formula.

However, since the universal quantification $\forall t$ commutes with the universal quantification on points of outcomes $\mathbf{AG}$ and with the binding $(Ag,\boldsymbol{s})$ (which does not involve t), one can instead rewrite Φ_{SPE} as follows:

$$\Phi''_{SPE} = \exists s_1 \dots \exists s_n\, \forall t\, (Ag,\boldsymbol{s})\mathbf{AG} \bigwedge_{i=1}^{n} (a_i,t)\psi_i \to (Ag,\boldsymbol{s})\psi_i,$$

This formula is in BSL[NG], *and even in its flat fragment.*

We conclude this section by recalling the usual semantics of BSL.

2.3 White-box Semantics of Strategy Logic

We recall the classical Tarski-style semantics of BSL formulas, which we call *white-box* and denote $\models_\square$. In the next section we will provide a new equivalent definition via evaluation games.

Definition 4 (White-box semantics)**.** *Let $\mathcal{G}$ be a game. A history formula φ (resp. outcome formula ψ) is evaluated in an assignment χ for free variables of φ (resp. ψ), a binding β and a history h (resp. a play π and an index $i \in \mathbb{N}$). We define $\mathcal{G},\chi,\beta,h \models_\square \varphi$ and $\mathcal{G},\chi,\beta,\pi,i \models_\square \psi$ inductively as follows (we omit $\mathcal{G}$):*

$$\begin{array}{lll}
\chi,\beta,h \models_\square p & \textit{if} & p \in \ell(\textit{last}(h))\\
\chi,\beta,h \models_\square \neg\varphi & \textit{if} & \chi,\beta,h \not\models_\square \neg\varphi\\
\chi,\beta,h \models_\square \varphi\vee\varphi' & \textit{if} & \chi,\beta,h \models_\square \varphi \textit{ or } \chi,\beta,h \models_\square \varphi'\\
\chi,\beta,h \models_\square \exists s\,\varphi & \textit{if} & \exists\sigma \in \textit{Str s.t. } \chi[s\mapsto\sigma],\beta,h \models_\square \varphi\\
\chi,\beta,h \models_\square (a,s)\varphi & \textit{if} & \chi,\beta[a\mapsto\chi(s)],h \models_\square \varphi\\
\chi,\beta,h \models_\square (a,\emptyset)\varphi & \textit{if} & \chi,\beta[a\mapsto\emptyset],h \models_\square \varphi\\
\chi,\beta,h \models_\square \mathbf{E}\psi & \textit{if} & \exists\pi \in \textit{Out}(h,\beta) \textit{ s.t.}\\
 & & \chi,\beta,\pi,|h| \models_\square \psi\\
\chi,\beta,\pi,i \models_\square \varphi & \textit{if} & \chi,\beta,\pi_{\le i} \models_\square \varphi\\
\chi,\beta,\pi,i \models_\square \neg\psi & \textit{if} & \chi,\beta,\pi,i \not\models_\square \psi\\
\chi,\beta,\pi,i \models_\square \psi\vee\psi' & \textit{if} & \chi,\beta,\pi,i \models_\square \psi \textit{ or } \chi,\beta,\pi,i \models \psi'\\
\chi,\beta,\pi,i \models_\square \mathbf{X}\psi & \textit{if} & \chi,\beta,\pi,i+1 \models_\square \psi\\
\chi,\beta,\pi,i \models_\square \psi\mathbf{U}\psi' & \textit{if} & \exists j \ge i \textit{ s.t. } \chi,\beta,\pi,j \models_\square \psi', \textit{ and}\\
 & & \forall k \in [i,j),\ \chi,\beta,\pi,k \models_\square \psi
\end{array}$$

For a sentence φ we write $\mathcal{G} \models_\square \varphi$ if $\mathcal{G},\emptyset,\emptyset,v_0 \models_\square \varphi$.

In this paper we focus on strategy quantification, and in particular alternation of existential and universal quantifiers. In the white-box semantics of a formula $\forall s.\exists s'.\varphi$ the strategy σ chosen for s is publicly announced, in the sense that s' can depend on σ. Technically this is due to the fact that the evaluation of $\exists s'.\varphi$ is carried out with σ in the assignment $\chi[s\mapsto\sigma]$. Let us illustrate this on a very simple example.

[1] which is implicit here, see Remark 1

Example 2. *"Cop and robber" is a one-step concurrent two-player game. Both the cop and the robber pick either* LEFT *or* RIGHT *at the same time, and the cop catches the robber if their choices match (see Figure 1).*

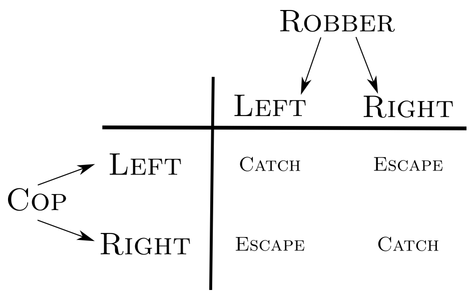

Figure 1: Cop and robber

Consider the BSL *formula*

$$\varphi = \forall s\, \exists t\, (\text{COP}, s)(\text{ROBBER}, t)\text{ESCAPE}$$

*It expresses that for every strategy of the cop there is a strategy of the robber which ensures that she escapes (*ESCAPE *is a formula which expresses that the outcome of the one-shot game is* ESCAPE*).*

In the white-box semantics, we have $\mathcal{G} \models_{\Box} \varphi$. Indeed, when the robber picks a strategy, the cop's strategy s has already been publicly announced, hence choosing $t \neq s$ ensures an escape. We will get back to this example with the black-box semantics.

3 Black-box Semantics

In this section we introduce and discuss the new black-box semantics, which we define via evaluation games. We first introduce an alternative definition of the usual white-box semantics via evaluation games with perfect information, and then describe how to obtain the black-box semantics by hiding the chosen strategies in the games for the white-box semantics, thus obtaining games with imperfect information. We refer to, e.g., (Hodges and Väänänen 2019) for an introduction to game semantics for classical logic.

3.1 White-box Evaluation Games

Given a game $\mathcal{G}$ and a BSL sentence Φ, we define the evaluation game $\mathbb{G}_{\Box}(\mathcal{G}, \Phi)$. It is a turn-based game played between two players: Eloise (also written E), who aims at proving that $\mathcal{G} \models_{\Box} \Phi$, and Abelard (A), who challenges this claim.

The set $V_{\mathcal{G},\Phi}$ of positions in $\mathbb{G}_{\Box}(\mathcal{G}, \Phi)$ is the union of

$$\{(\chi, \beta, h, \varphi, \text{P}) \mid \varphi \in \text{HistSub}(\Phi), \text{P} \in \{\text{E}, \text{A}\}\}$$

and

$$\{(\chi, \beta, \pi, i, \psi, \text{P}) \mid \psi \in \text{OutSub}(\Phi), \text{P} \in \{\text{E}, \text{A}\}\}$$

and the initial position is $(\emptyset, \emptyset, v_0, \Phi, V)$. Negation is dealt with by switching the roles of the players in trying to prove or disprove the current subformula, the idea being that Eloise succeeds in proving that $\neg\varphi$ holds if Abelard fails to prove that φ holds. Component $\text{P} \in \{\text{E}, \text{A}\}$ in a position $(\chi, \beta, h, \varphi, \text{P})$ indicates who of Eloise or Abelard is trying to prove subformula φ true in the current context; this is determined by whether the number of negations above subformula φ is even or odd[2]. We call P the *player in charge* and let $\overline{\text{P}}$ denote its opponent.

Positions of the form $(\chi, \beta, h, p, \text{P})$ are terminal, meaning that the game ends. Such a position with $\text{P} = \text{E}$ is *winning* for Eloise if $p \in \ell(\text{last}(h))$, otherwise it is winning for Abelard, and vice versa when $\text{P} = \text{A}$. Moves in other types of positions are as follows.

- $\neg$-positions: from $(\chi, \beta, h, \neg\varphi, \text{P})$ the game goes to $(\chi, \beta, h, \varphi, \overline{\text{P}})$, and similarly from $(\chi, \beta, \pi, i, \neg\psi, \text{P})$ the game goes to $(\chi, \beta, \pi, i, \psi, \overline{\text{P}})$.
- $\vee$-positions: from $(\chi, \beta, h, \varphi \vee \varphi', \text{P})$, P moves either to $(\chi, \beta, h, \varphi, \text{P})$ or to $(\chi, \beta, h, \varphi', \text{P})$. Similarly for positions of the form $(\chi, \beta, \pi, i, \psi \vee \psi', \text{P})$.
- $\exists s$-positions: from $(\chi, \beta, h, \exists s\, \varphi, \text{P})$, P chooses some $\sigma \in \text{Str}$ and moves to $(\chi[s \mapsto \sigma], \beta, h, \varphi, \text{P})$.
- (a, s)-positions: from $(\chi, \beta, h, (a, s)\varphi, \text{P})$, the game moves to $(\chi, \beta[a \mapsto \chi(s)], h, \varphi, \text{P})$.
- $(a, \emptyset)$-positions: from $(\chi, \beta, h, (a, s)\varphi, \text{P})$, the game moves to $(\chi, \beta[a \mapsto \emptyset], h, \varphi, \text{P})$.
- $\mathbf{E}$-positions: from $(\chi, \beta, h, \mathbf{E}\psi, \text{P})$, P chooses some outcome $\pi \in \text{Out}(\chi, h)$ and moves to $(\chi, \beta, \pi, |h|, \varphi, 0)$.
- φ-positions: from $(\chi, \beta, \pi, i, \varphi, \text{P})$ the game goes to $(\chi, \beta, \pi_{\leq i}, \varphi, \text{P})$
- $\mathbf{X}$-positions: from $(\chi, \beta, \pi, i, \mathbf{X}\psi, \text{P})$, the game moves to $(\chi, \beta, \pi, i+1, \psi, \text{P})$.
- $\mathbf{U}$-positions: from $(\chi, \beta, \pi, i, \psi \mathbf{U} \psi', \text{P})$, P chooses between $(\chi, \beta, \pi, i, \psi', \text{P})$ and $(\chi, \beta, \pi, i, \psi \wedge \mathbf{X}\psi\mathbf{U}\psi', \text{P})$.

A play is *winning* if either it reaches a winning terminal position, or it remains from some point onwards in $\mathbf{U}$-positions with $\text{P} = \text{A}$ (meaning that Abelard is trying to prove true some formula $\psi\mathbf{U}\psi'$ but keeps postponing forever the realization of ψ').

A *strategy* for Eloise defines a move for every history in $\mathbb{G}_{\Box}(\mathcal{G}, \Phi)$ ending in a situation where Eloise has a choice to make. A strategy for Eloise is *winning* if all plays consistent with this strategy are winning.

Lemma 1 (Correctness of the evaluation game)**.** *$\mathcal{G} \models_{\Box} \Phi$ if, and only if, Eloise has a winning strategy in $\mathbb{G}_{\Box}(\mathcal{G}, \Phi)$ from the initial position.*

3.2 Black-box Evaluation Games

The intuition behind the black-box semantics is that when a strategy is universally quantified, it is fixed but not publicly announced, so that later existentially quantified strategies cannot depend on them. This is mathematically formalised by constraining the information available to Eloise in the evaluation game, making it an imperfect-information game in the classical sense for games played on graphs, where an

[2]Note that all along Eloise tries to prove that the initial formula holds in the game. Roles change only for subformulas: Eloise proves that $\neg\varphi$ is true if Abelard cannot prove that φ is true.

equivalence relation models indistinguishable positions, and strategies are required to be consistent with observations, i.e., they should assign the same actions to equivalent histories (Doyen and Raskin 2011).

We define the fact that a sentence Φ of BSL is satisfied in a game $\mathcal{G}$ with the black-box semantics, written $\mathcal{G} \models_{\blacksquare} \Phi$, via an evaluation game $\mathbb{G}_{\blacksquare}(\mathcal{G}, \Phi)$.

The game $\mathbb{G}_{\blacksquare}(\mathcal{G}, \Phi)$ is similar to $\mathbb{G}_{\square}(\mathcal{G}, \Phi)$ (it has the same set of positions $V_{\mathcal{G},\varphi}$, same moves and same winning condition), except that it has imperfect information: we define an equivalence relation $\sim$ over $V_{\mathcal{G},\Phi}$ by letting two positions be equivalent if they are identical except possibly for the interpretation of universally quantified variables. We then require that Eloise play similarly in equivalent situations, thus forcing her to choose existentially quantified strategies independently from the universally quantified strategies chosen by Abelard.

Formally, define $(\chi, \beta, h, \varphi, \mathrm{P}) \sim (\chi', \beta', h', \varphi', \mathrm{P}')$ if $\beta = \beta'$, $h = h'$, $\varphi = \varphi'$, $\mathrm{P} = \mathrm{P}'$ and for all $s \in \mathrm{Var}$ existentially quantified in Φ, we have $\chi(s) = \chi'(s)$. Similarly, define $(\chi, \beta, \pi, i, \psi, \mathrm{P}) \sim (\chi', \beta', \pi', i', \psi', \mathrm{P}')$ if $\beta = \beta'$, $\pi = \pi'$, $i = i'$, $\psi = \psi'$, $\mathrm{P} = \mathrm{P}'$ and for all $s \in \mathrm{Var}$ existentially quantified in Φ, we have $\chi(s) = \chi'(s)$. We then let $[V_{\mathcal{G},\Phi}]_{\sim}$ denote the quotient of $V_{\mathcal{G},\Phi}$ by $\sim$, and require that Eloise's strategies be defined on $[V_{\mathcal{G},\Phi}]^*_{\sim}$ instead of $V^*_{\mathcal{G},\Phi}$ as in $\mathbb{G}_{\square}(\mathcal{G}, \Phi)$.

Definition 5 (Black-box semantics)**.** *Let $\mathcal{G}$ be a game, Φ a* BSL *sentence and φ a history subformula of Φ. Then $\mathcal{G}, \chi, \beta, h \models_{\blacksquare} \varphi$ if Eloise has a winning strategy in $\mathbb{G}_{\blacksquare}(\mathcal{G}, \Phi)$ from position $(\chi, \beta, h, \varphi, \mathrm{E})$.*

We start by proving the following central property, which is that under black-box semantics existential and universal strategy quantifiers commute. The proof also serves to illustrate how the game semantics works.

Lemma 2. *For every formula of the form $\forall s\, \exists t\, \Phi$, for every game $\mathcal{G}$, assignment χ, binding β and history h, it holds that*

$$\mathcal{G}, \chi, \beta, h \models_{\blacksquare} \forall s\, \exists t\, \Phi \quad \textit{iff} \quad \mathcal{G}, \chi, \beta, h \models_{\blacksquare} \exists t\, \forall s\, \Phi$$

Proof. The right-to-left direction is clear. The other one follows directly from the definition of the black-box semantics: assume that $\mathcal{G}, \chi, \beta, h \models_{\blacksquare} \forall s\, \exists t\, \Phi$, i.e., Eloise has a winning strategy in the corresponding evaluation game from position $(\chi, \beta, h, \neg\exists s\, \neg\exists t\, \Phi, \mathrm{E})$. The game moves to $(\chi, \beta, h, \exists s\, \neg\exists t\, \Phi, \mathrm{A})$, where Abelard chooses a strategy σ, then to $(\chi[s \mapsto \sigma], \beta, h, \neg\exists t\, \Phi, \mathrm{A})$, and then to $(\chi[s \mapsto \sigma], \beta, h, \exists t\, \Phi, \mathrm{E})$. Let σ' be the strategy chosen by Eloise's winning strategy in this position. By definition of the imperfect-information evaluation game, the strategy σ' picked by Eloise's winning strategy is the same for any choice made by Abelard for t, and the position $(\chi[s \mapsto \sigma, t \mapsto \sigma'], \beta, h, \Phi, \mathrm{E})$ is winning for Eloise for any σ. As a result Eloise has a winning strategy from $(\chi, \beta, h, \exists t\, \neg\exists s\, \neg\Phi, \mathrm{E})$, which is to pick σ', and this concludes the proof. $\square$

Example 3. *Let us get back to Example 2. Intuitively, if the cop keeps its strategy private, then there is no way for the robber to pick a strategy that ensures she escapes. This is the kind of situation that the black-box semantics is designed to capture, and we claim that indeed $\mathcal{G} \not\models_{\blacksquare} \varphi$. This is because a strategy for Eloise for the subformula $\psi = \exists t\, (\mathrm{Cop}, s)(\mathrm{Robber}, t)$ Escape does not see what strategy Abelard has chosen for s. In other words, it must make the same choice for all s, but there is no strategy t for the* Robber *that ensures she will escape for any s.*

Example 4. *The point of this example is to illustrate a subtelty of the black-box semantics for existential quantifiers. We consider a variant of the cop and robber game from Example 2: now there are two cops,* Cop_1 *and* Cop_2*, and one robber. All three agents choose* Left *or* Right *at the same time, and the cops catch the robber if at least one of the two cops chooses the same direction as the robber.*

Consider the SL *formula*

$$\exists s_1\, \exists s_2\, \forall t\, (\mathrm{Cop}_1, s_1)(\mathrm{Cop}_2, s_2)(\mathrm{Robber}, t)\ \mathrm{Catch}$$

It expresses that there exist strategies for the two cops ensuring to catch the robber, and intuitively it should hold: two strategies s_1 and s_2 that pick different directions are witnesses for the satisfaction of this formula. And indeed we have that $\mathcal{G} \models_{\blacksquare} \Phi$.

There are two aspects of the evaluation game that account for this. First, only the valuation of universally *quantified strategies is hidden, while existential ones are visible to Eloise. Second, all existentially quantified strategies are chosen by Eloise, and that a formula is said to hold if there exists a winning strategy for Eloise. This meta existential quantification on Eloise's strategies effectively quantifies on all existentially quantified strategies of the formula at once, making it possible for s_2 to depend on s_1. A consequence of this is that the semantics would remain unchanged if all strategies (both existentially and universally quantified) were hidden in the evaluation game $\mathbb{G}_{\blacksquare}(\mathcal{G}, \Phi)$.*

Example 5. *Consider again formulas Φ_{NE} and Φ''_{SPE} from Example 1. In both formulas, all existential quantifications on strategies are made* before *universal ones, so that black-box and white-box semantics coincide for these formulas.*

From the last example we get:

Proposition 1. *The existence of Nash Equilibria and Subgame Perfect Equilibria can be expressed in* BSL[NG] *with black-box semantics.*

3.3 Comparison with ATL*

We briefly recall the syntax and semantics of ATL*.

Formulas are given by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle A \rangle \psi$$
$$\psi ::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\psi \mid \psi \mathbf{U} \psi$$

where $p \in \mathrm{AP}$ and $A \subseteq \mathrm{Ag}$.

A formula of type φ is evaluated in a position of a game, and a formula of type ψ is evaluated in a play. We only recall the semantics for the coalition operator $\langle A \rangle \psi$, whose intuitive meaning is that coalition A has a strategy to ensure that ψ holds: $\mathcal{G}, v \models \langle A \rangle \psi$ holds if there exists a strategy profile $\{\sigma_a\}_{a \in A}$ such that for all outcomes $\pi \in \mathrm{Out}(v, (a, \sigma_a)_{a \in A})$ formula ψ holds in π. The dual operator $[A]\psi$ is defined as $\neg\langle A\rangle\neg\psi$, and means that for any strategy of coalition A there exists an outcome that satisfies ψ.

ATL^* formulas can be translated to BSL with black-box semantics. Such a translation can be defined by induction, where all cases are trivial but the coalition quantifier which is as follows: assuming that $\mathrm{Ag} = \{a_1, \ldots, a_n\}$ and $A = \{a_{i_1}, \ldots, a_{i_k}\}$, $\langle A\rangle\psi$ is translated as

$$\exists s_1 \ldots \exists s_k(\mathrm{Ag}, \emptyset)(a_1, s_1) \ldots (a_k, s_k)\mathbf{A}\psi',$$

where ψ' is the translation of ψ.

Observe that each block of strategy quantifiers is followed by a unique goal which starts by unbinding all agents, and all new bindings are made with newly quantified strategies. As a result, this translation produces only formulas in the Nested Goals fragment. In addition, each block of strategy quantifiers consists in only existential operators (when translating $\langle A\rangle$) or only universal ones (when translating $[A]$), so that the white-box and black-box semantics coincide for the obtained formulas.

It follows that $\mathsf{BSL}[\mathsf{NG}]$ with black-box semantics subsumes ATL^*. We show that it is actually strictly more expressive.

Proposition 2. *$\mathsf{BSL}[\mathsf{NG}]$ with black-box semantics is strictly more expressive than ATL^*.*

Proof. We adapt the proof from (Mogavero et al. 2014) that One-Goal SL with white-box semantics is strictly more expressive than ATL^*. Consider the two concurrent game structures in Figure 2. We have three agents a, b and c, and two actions 0 and 1 in $\mathcal{G}_1$, three actions 0, 1 and 2 in $\mathcal{G}_2$. In both games the initial position is v_0. The transitions are defined by the sets of joint moves $J_1 = \{00*, 11*\}$ and $J_2 = \{00*, 11*, 12*, 200, 202, 211\}$ ($\overline{J_i}$ denotes the complement of J_i). It is proved in (Mogavero et al. 2014, Theorem 4.4) that both games satisfy the same ATL^* formulas.

Consider now the $\mathsf{BSL}[\mathsf{NG}]$ formula

$$\Phi = \exists s_a \forall s_b\, (a, s_a)(b, s_b)\mathbf{EX}\neg p$$

It is quite easy to see that $\mathcal{G}_1 \not\models_{\blacksquare} \Phi$, while $\mathcal{G}_2 \models_{\blacksquare} \Phi$. Indeed in $\mathcal{G}_1$ no matter which action is chosen by a, if b plays the same action then the game goes to v_1, where p holds. In $\mathcal{G}_2$ instead, if agent a plays 2, then no matter what strategy is played by b, c has a way to reach v_2 where p does not hold. Observe that in Φ there is no universal quantifier under an existential one, so that white-box and black-box semantics coincide for this formula. $\square$

4 Model Checking

In this section we show that model checking $\mathsf{BSL}[\mathsf{NG}]$ with black-box semantics is 3-EXPTIME-complete.

4.1 Upper Bound

We establish the result for the flat fragment. It can then be lifted to $\mathsf{BSL}[\mathsf{NG}]$ via a marking algorithm that evaluates independent subsentences in a bottom-up fashion. But first we recall the notion of alternation depth of a formula.

Definition 6. *The alternation depth of a formula in $\mathsf{BSL}[\mathsf{NG,f}]$ is the maximum number of alternations between existential and universal strategy quantifiers.*

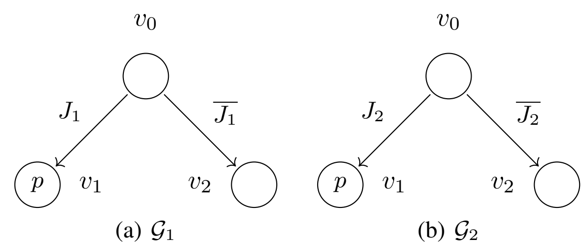

Figure 2: Games $\mathcal{G}_1$ and $\mathcal{G}_2$

We now establish the following about the flat fragment:

Proposition 3. *Model checking $\mathsf{BSL}[\mathsf{NG,f}]$ with black-box semantics is in 3-EXPTIME.*

Proof. Let Φ be a formula in the flat fragment of $\mathsf{BSL}[\mathsf{NG}]$. It is of the form $\Phi = Q\phi$, where Q is a block of strategy quantifiers and ϕ contains no strategy quantifier. By Lemma 2, existential quantifiers in Q can be permuted with universal ones, thus obtaining an equivalent formula $\Phi' = Q'\phi$ where Q' consists of a block of existential strategy quantifiers followed by a block of universal ones. Φ' has alternation depth 1 and model checking BSL formulas with alternation depth k can be done in time at most $(k+2)$-exponential in the size of the formula (Berthon et al. 2021, Proposition 5.4).[3] $\square$

Note that formulas obtained by the translation from ATL^* to $\mathsf{BSL}[\mathsf{NG,f}]$ presented in Section 3.3 have zero alternation of strategy quantifiers, so that they fall in a fragment of $\mathsf{BSL}[\mathsf{NG,f}]$ whose model checking is actually in 2-EXPTIME, as for ATL^*.

4.2 Lower Bound

We prove the following:

Proposition 4. *Model checking $\mathsf{BSL}[\mathsf{NG,f}]$ with black-box semantics is 3-EXPTIME-hard.*

Proof. It is shown in (Laroussinie and Markey 2014, Theorem 4.11) that model checking $\mathsf{EQ}^k\mathsf{CTL}^*$ is $(k+1)$-EXPTIME hard for $k > 0$, where $\mathsf{EQ}^k\mathsf{CTL}^*$ is the fragment of Quantified CTL^* with all second-order quantifiers at the beginning of the formula, starting with an existential quantification and with $k-1$ alternations. The reduction from model checking for Quantified CTL^* to that of BSL with white-box semantics presented in (Laroussinie and Markey 2015; Berthon et al. 2021), when applied to formulas of $\mathsf{EQ}^k\mathsf{CTL}^*$, produces formulas in $\mathsf{BSL}[\mathsf{NG,f}]$ with alternation $k-1$. Taking $k = 2$, we obtain formulas in which the

[3]More precisely (Berthon et al. 2021) considers a notion of *simulation depth* which allows for a finer analysis, and they prove that formulas with simulation depth k can be model checked in $(k+1)$-exponential time. Formulas of alternation depth k as defined here have simulation depth at most $k+1$, hence the result. Some formulas of alternation k have simulation depth only k, but defining simulation depth would be cumbersome and unnecessary to obtain our 3-EXPTIME upper bound.

only strategy quantifiers consist of an initial block of existential quantifiers followed by a block of universal ones. For such formulas, the black-box semantics coincides with the white-box one. We thus obtain the desired lower bound. □

Propositions 3 and 4 entail that:

Theorem 1. *Model checking* BSL[NG] *with black-box semantics is* 3*-EXPTIME-complete.*

Up to our knowledge, BSL[NG] is the first logic that strictly subsumes ATL^*, can express Nash equilibria and subgame perfect Equilibria, and yet enjoys a model-checking problem with elementary complexity.

5 Strategy Logic with Mixed Announcements

We now introduce a logic that combines the white-box and black-box semantics, by allowing two kinds of strategy quantification: public (white-box) and private (black-box). This logic subsumes both BSL with white-box semantics and BSL with black-box semantics, and is written $\mathsf{BSL}_{\square\blacksquare}$.

Definition 7 (Syntax)**.** *The set of* $\mathsf{BSL}_{\square\blacksquare}$ *formulas is given by the following grammar:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists^{\square} s\, \varphi \mid \exists^{\blacksquare} s\, \varphi \mid (a, s)\varphi \mid (a, \emptyset)\varphi \mid \mathbf{E}\psi$$
$$\psi ::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\psi \mid \psi\mathbf{U}\psi$$

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

Define $\forall^{\square} s\, \varphi = \neg\exists^{\square}\neg\varphi$ and $\forall^{\blacksquare} s\, \varphi = \neg\exists^{\blacksquare}\neg\varphi$. Strategy variables introduced by white-box quantifiers $\exists^{\square} s$ are called *publicly quantified*, while those introduced by black-box quantifiers $\exists^{\blacksquare} s$ are called *privately quantified*.

We now define the semantics $\models_{\square\blacksquare}$ of $\mathsf{BSL}_{\square\blacksquare}$ via yet another evaluation game, obtained by refining the information accessible to Eloise in the black-box evaluation games. More precisely, for a game $\mathcal{G}$ and a sentence Φ we define the evaluation game $\mathbb{G}_{\square\blacksquare}(\mathcal{G}, \Phi)$ as follows: the set of positions $V^{\square\blacksquare}_{\mathcal{G},\Phi}$ is again the union of

$$\{(\chi, \beta, h, \varphi, \mathrm{P}) \mid \varphi \in \mathrm{HistSub}(\Phi), \mathrm{P} \in \{\mathrm{E}, \mathrm{A}\}\} \text{ and}$$
$$\{(\chi, \beta, \pi, i, \psi, \mathrm{P}) \mid \psi \in \mathrm{OutSub}(\Phi), \mathrm{P} \in \{\mathrm{E}, \mathrm{A}\}\},$$

the initial position is $(\emptyset, \emptyset, v_0, \Phi, V)$, and moves are defined similarly to the games for white-box and black-box semantics. In particular, for the two types of strategy quantifiers, moves are as follows:

- $\exists^{\square} s$-positions: from $(\chi, \beta, h, \exists^{\square} s\, \varphi, \mathrm{P})$, P chooses some $\sigma \in \mathrm{Str}$ and moves to $(\chi[s \mapsto \sigma], \beta, h, \varphi, \mathrm{P})$.
- $\exists^{\blacksquare} s$-positions: from $(\chi, \beta, h, \exists^{\blacksquare} s\, \varphi, \mathrm{P})$, P chooses some $\sigma \in \mathrm{Str}$ and moves to $(\chi[s \mapsto \sigma], \beta, h, \varphi, \mathrm{P})$.

The only difference is in the definition of the equivalence relation on positions. Formally, we define $(\chi, \beta, h, \varphi, \mathrm{P}) \sim_{\square\blacksquare} (\chi', \beta', h', \varphi', \mathrm{P}')$ if $\beta = \beta'$, $h = h'$, $\varphi = \varphi'$, $\mathrm{P} = \mathrm{P}'$ and for all variables $s \in \mathrm{Var}$ that are either existentially quantified or universally *publicly* quantified in Φ, we have $\chi(s) = \chi'(s)$; similarly for $(\chi, \beta, \pi, i, \psi, \mathrm{P}) \sim_{\square\blacksquare} (\chi', \beta', \pi', i', \psi', \mathrm{P}')$. Strategies of Eloise are defined on $[V^{\square\blacksquare}_{\mathcal{G},\Phi}]^*_{\sim_{\square\blacksquare}}$. The interpretation is that Eloise cannot observe which strategies are chosen by Abelard for universal black-box quantifiers.

Definition 8 (Mixed announcements semantics)**.** *Let* $\mathcal{G}$ *be a game,* Φ *a* $\mathsf{BSL}_{\square\blacksquare}$ *sentence and* φ *a history subformula of* Φ*. Then* $\mathcal{G}, \chi, \beta, h \models_{\square\blacksquare} \varphi$ *if Eloise has a winning strategy in* $\mathbb{G}_{\square\blacksquare}(\mathcal{G}, \Phi)$ *from position* $(\chi, \beta, h, \varphi, \mathrm{E})$*.*

First, note that by definition of the semantics, and in particular by definition of $\sim_{\square\blacksquare}$, public or private quantification makes a difference only for *universal* quantification, so that we may write $\exists s$ instead of $\exists^{\square} s$ or $\exists^{\blacksquare} s$.

We now illustrate with two examples how BSL with mixed announcements can be used to model subtle scenarios in which agents in a system may or may not be informed of other agents' strategies.

Example 6. *Let us discuss another variant of the cop and robber game from Example 2 with two cops and one robber. Consider the* $\mathsf{BSL}_{\square\blacksquare}$ *formula:*

$$\Phi = \forall^{\blacksquare} s_1 \forall^{\blacksquare} t\, \exists s_2 (\textsc{Cop}_1, s_1)(\textsc{Cop}_2, s_2)(\textsc{Robber}, t)\textsc{Catch}$$

It expresses that the second cop can choose a strategy s_2 *ensuring a catch irrespective of both the strategies of the first cop* (s_1) *and of the robber* (t)*. Note that here it is quite natural to require that* s_2 *does not depend on* t*, which is why* t *is quantified privately. Whether* s_2 *should be allowed to depend on* s_1 *or not depends on whether the two cops can communicate along a private channel, and the point of this example is to show that this can be formalised in* BSL *with mixed announcements.*

The formula above does not hold: $\mathcal{G} \not\models_{\square\blacksquare} \Phi$*. Indeed, to ensure a catch,* s_2 *should choose the direction not chosen by* s_2*. But since* s_1 *is kept private,* s_2 *cannot depend on it. However, by replacing* $\forall^{\blacksquare} s_1$ *with* $\forall^{\square} s_1$*, the formula becomes true on this game, because* s_2 *can make the opposite choice of* s_1 *to ensure a catch.*

Example 7. *All examples so far were one-step concurrent games. To illustrate the expressive power of strategy logic with mixed announcements, let us consider a multiagent system involving temporal reasoning. The game is called "cop and smuggler": at each time step, the smuggler chooses either* Left*,* Right*, or* Wait*, and the cop chooses either* Left *or* Right*. If at some point the two choices coincide, the cop catches the robber. The outcome is illustrated in Figure 3. The objective of the smuggler is to smuggle as much as possible.*

		Smuggler		
		Left	Wait	Right
Cop	Left	Catch	—	Smuggle
	Right	Smuggle	—	Catch

Figure 3: Cop and smuggler

Consider the $\mathsf{BSL}_{\square\blacksquare}$ *formula:*

$$\forall^{\square} t\, (\textsc{Smuggler}, t) \Big(\mathbf{AGF}\, (\textsc{Left} \vee \textsc{Right}) \rightarrow \exists s\, (\textsc{Cop}, s)\, \mathbf{F}\, \textsc{Catch}\Big)$$

It expresses that for all strategies of the robber that attempts to smuggle infinitely often, i.e., does not wait forever, there exists a catching strategy for the cop. The formula holds if t *is publicly announced (as in the formula above), and it does not otherwise.*

6 Conclusion

Strategy logic with black-box semantics is a natural logic for reasoning about agent-based scenarios in which agents do not announce their strategies. Its Nested-Goals fragment is situated in a middle ground between the expressiveness of SL with white-box semantics and the elementary complexity of ATL^*. Indeed, similarly to SL with white-box semantics, it can express certain complex solution concepts such as Nash equilibria. On the other hand, similarly to ATL^*, it has an elementary complexity, i.e., a 3-EXPTIME-complete model-checking problem, which is much closer to ATL^* (2-EXPTIME-complete) than SL with white-box semantics (non-elementary).

The behavioural semantics discussed in the related works, which also aims at obtaining a natural and expressive semantics for SL with good computational complexity, is defined only for the Nested-Goals fragment and relies on complex dependency maps. In contrast our black-box semantics is defined for the whole logic BSL via intuitive evaluation games with imperfect information.

One direction for future work is to establish the complexity of model checking the full BSL with black-box semantics, as well as that of the logic with mixed announcements $\mathsf{BSL}_{\square\blacksquare}$. Another would be to draw inspiration from independence-friendly logic (Mann, Sandu, and Sevenster 2011) and dependence logic (Väänänen 2007) to define a *team semantics* for Strategy Logic with private announcements. Indeed these logics study mechanisms of dependence among variables in classical logic, which is closely related to what we study here. Dependence logic considers semantics based on imperfect-information games, as we do, but also has a Tarski-style semantics based on *teams*: intuitively a formula is not evaluated with respect to a single assignment for free variables, but a set of assignments, which allows formalizing dependencies between variables.

References

Alur, R.; Henzinger, T. A.; and Kupferman, O. 2002. Alternating-time temporal logic. *J. ACM* 49(5):672–713.

Aminof, B.; Kwiatkowska, M.; Maubert, B.; Murano, A.; and Rubin, S. 2019. Probabilistic strategy logic. In Kraus, S., ed., *IJCAI'19*, 32–38. ijcai.org.

Belardinelli, F.; Jamroga, W.; Kurpiewski, D.; Malvone, V.; and Murano, A. 2019. Strategy logic with simple goals: Tractable reasoning about strategies. In Kraus, S., ed., *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 88–94. ijcai.org.

Berthon, R.; Maubert, B.; Murano, A.; Rubin, S.; and Vardi, M. Y. 2021. Strategy logic with imperfect information. *ACM Trans. Comput. Log.* 22(1):5:1–5:51.

Čermák, P.; Lomuscio, A.; Mogavero, F.; and Murano, A. 2018. Practical verification of multi-agent systems against slk specifications. *Inf. Comput.* 261:588–614.

Cermák, P.; Lomuscio, A.; and Murano, A. 2015. Verifying and synthesising multi-agent systems against one-goal strategy logic specifications. In Bonet, B., and Koenig, S., eds., *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, 2038–2044. AAAI Press.

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

Doyen, L., and Raskin, J.-F. 2011. Games with imperfect information: Theory and algorithms. In *Lectures in Game Theory for Computer Scientists*. 185–212.

Fijalkow, N.; Maubert, B.; Murano, A.; and Rubin, S. 2018. Quantifying bounds in strategy logic. In *CSL'18*, volume 23.

Gardy, P.; Bouyer, P.; and Markey, N. 2020. Dependences in strategy logic. *Theory Comput. Syst.* 64(3):467–507.

Henkin, L., and Karp, C. R. 1965. Some remarks on infinitely long formulas. *Journal of Symbolic Logic* 30(1).

Hintikka, J., and Sandu, G. 1997. Game-theoretical semantics. In van Benthem, J., and ter Meulen, A., eds., *Handbook of Logic and Language*. North Holland / Elsevier. 361–410.

Hintikka, J. 1998. Informational independence as a semantical phenomenon. In *Paradigms for Language Theory and Other Essays*. Springer. 52–70.

Hodges, W., and Väänänen, J. 2019. Logic and Games. In Zalta, E. N., ed., *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2019 edition.

Knight, S., and Maubert, B. 2015. Dealing with imperfect information in strategy logic. In *SR'15*.

Laroussinie, F., and Markey, N. 2013. Satisfiability of ATL with strategy contexts. In *GandALF'13*, 208–223.

Laroussinie, F., and Markey, N. 2014. Quantified CTL: expressiveness and complexity. *Log. Methods Comput. Sci.* 10(4).

Laroussinie, F., and Markey, N. 2015. Augmenting ATL with strategy contexts. *Inf. Comput.* 245:98–123.

Mann, A. L.; Sandu, G.; and Sevenster, M. 2011. *Independence-friendly logic: A game-theoretic approach*, volume 386. Cambridge University Press.

Mogavero, F.; Murano, A.; Perelli, G.; and Vardi, M. Y. 2014. Reasoning about strategies: On the model-checking problem. *ACM Trans. Comput. Log.* 15(4):34:1–34:47.

Mogavero, F.; Murano, A.; Perelli, G.; and Vardi, M. 2017. Reasoning about strategies: on the satisfiability problem. *Logical Methods in Computer Science* 13(1).

Pnueli, A., and Rosner, R. 1989. On the synthesis of a reactive module. In *POPL'89*, 179–190.

Väänänen, J. 2007. *Dependence logic: A new approach to independence friendly logic*, volume 70. Cambridge University Press.