Reasoning About Agents That May Know Other Agents’ Strategies

Francesco Belardinelli¹, Sophia Knight², Alessio Lomuscio¹, Bastien Maubert³, Aniello Murano³ and Sasha Rubin⁴

¹Imperial College London, UK
²University of Minnesota Duluth, USA
³Università degli Studi di Napoli “Federico II”, Italy
⁴University of Sydney, Australia

Abstract

We study the semantics of knowledge in strategic reasoning. Most existing works either implicitly assume that agents do not know one another’s strategies, or that all strategies are known to all; and some works present inconsistent mixes of both features. We put forward a novel semantics for Strategy Logic with Knowledge that cleanly models whose strategies each agent knows. We study how adopting this semantics impacts agents’ knowledge and strategic ability, as well as the complexity of the model-checking problem.

1 Introduction

Reasoning about strategic abilities of agents and groups of agents is a topic of great interest in the formal study of multi-agent systems. To that end many logics have been introduced, such as Game Logic [Pauly and Parikh, 2003], STIT [Herzig and Lorini, 2010], and ATL [Alur et al., 2002].

Among these logics, Strategy Logic (SL) [Chatterjee et al., 2010; Mogavero et al., 2014] has proved very successful in recent years, as it combines a natural syntax with high expressive power that allow it to easily express complex game-theoretic notions such as existence of Nash equilibria. Its model checking problem is decidable, and model-checking algorithms can usually be used to synthesize strategies that satisfy a given specification. Because imperfect information is a crucial aspect of multi-agent systems, SL has recently been extended to take into account imperfect information [Berthon et al., 2021] and allow for epistemic reasoning [Maubert and Murano, 2018; Belardinelli et al., 2020].

As discussed in [Maubert and Murano, 2018], defining the semantics of knowledge in a strategic context involves subtleties with far-reaching consequences, that have often gone unheeded. Two semantics exist in the literature, and most works adopt one or the other without mentioning it. One corresponds to agents that do not know each other’s strategies (called uninformed semantics in [Maubert and Murano, 2018]), while the other corresponds to agents who know everyone’s strategies (informed semantics). The former is used in all existing epistemic extensions of ATL and SL (e.g., [van der Hoek and Wooldridge, 2003; Jamroga and van der Hoek, 2004; Guelev et al., 2011; Belardinelli et al., 2017b]), while the latter is adopted in works on distributed synthesis [van der Meyden and Wilke, 2005] and knowledge-based policies in POMDPs [Saffidine et al., 2018].

Intuitively, the uninformed semantics corresponds to situations where agents do not collaborate, or cannot communicate information about their strategies, and have no a priori knowledge of how other agents behave. On the other hand, the informed semantics corresponds to situations where the strategies followed by each agent are common knowledge. Knowing others’ strategies gives additional power to agents, as it allows them to infer more information than what they learn from their observations alone. One can think for instance of the Hanabi card game, where by knowing how other players choose their actions, one can interpret them and infer valuable information. However, this refined notion of knowledge comes at a price in terms of computational complexity.

Indeed some problems are decidable with the uninformed semantics and undecidable with the informed one. This is the case for distributed synthesis. This central problem in multi-agent systems is, in general, undecidable, already for simple reachability objectives without epistemic content [Petersen et al., 2001]. But it becomes decidable if there is a hierarchy of information among agents, or when all actions are observed similarly by all agents (see [Berthon et al., 2021] for a discussion). When objectives are no longer purely temporal but can also refer to knowledge, distributed synthesis for systems with hierarchical information remains decidable with the uninformed semantics of knowledge [Puchala, 2010], but it becomes undecidable for the informed semantics [van der Meyden and Wilke, 2005]. This gives even more reason to consider carefully, when studying a system, the assumptions made on the knowledge agents have of each other’s strategies.

Part of these results have been generalized to logics for strategic reasoning that capture distributed synthesis, but only for the uninformed semantics. Decidability has been generalized to an epistemic extension of SL in the case of hierarchical information [Maubert and Murano, 2018], and decidability in the case of public actions has been generalized to epistemic extensions of ATL [Belardinelli et al., 2017b] and SL [Belardinelli et al., 2020]. However no result exists for the informed semantics which, as far as we know, has never been formalized in the context of logics for strategic reasoning.

Contribution. We address this issue and propose a novel semantics of SL with imperfect information and epistemic
operators (SLK) in which one can specify, for each agent, the set of agents whose strategy she is informed of. This allows us to capture the case where all agents know everyone’s strategies (informed semantics) and the case where no one knows anyone else’s strategy (uninformed semantics), as well as intermediate situations (for instance, some agents may be allowed to privately communicate their strategies). We illustrate in detail the impact that knowing others’ strategies has on the agents’ knowledge as well as on their strategizing power. Doing so, we also shed light on an inconsistency in existing semantics of SL with knowledge, which is solved by our semantics. We then show that for the new semantics, model checking SLK is already undecidable under hierarchical information, but is decidable when all actions are public.

2 SLK with Informed Semantics

Fix AP a finite set of atomic propositions, Ag a finite set of agents, and Var a finite set of variables.

2.1 Syntax

We consider the flat fragment of SL with Boolean Goals [Mogavero et al., 2014] extended with knowledge operators. This simple syntax allows us to focus on the most important aspects of the new semantics, avoiding technical issues that would arise with the full syntax of SL and diminish the clarity of the exposition (see Remark 2 for more detail).

Definition 1. The syntax of SLK_Inf[BG] is given by the following grammar:

\[
\begin{align*}
\Phi & = s, \Phi | \lnot \Phi | \Phi \lor \Phi | \theta \\
\theta & = (Ag, s) \rho | \lnot \theta | \theta \land \theta \\
\varphi & = p | \lnot \varphi | \varphi \lor \varphi | X \varphi | \varphi U \varphi | K_a \varphi
\end{align*}
\]

where \( p \in AP, s \in Var, s \in Ag \rightarrow Var \times \{ \langle \rangle, [] \} \) and \( a \in Ag \).

\( X \) and \( U \) are the temporal operators “next” and “until”, \( \exists s \) is the existential quantifier on strategies, and \( K_a \) is the individual knowledge operator for agent \( a \). The group binding (\( (Ag, s) \)) assigns to each agent a strategy variable and a symbol indicating how the uniformity constraint should be treated. Intuitively, \( \langle \rangle \) (resp. \( [] \)) indicates a strategy that was quantified existentially (resp. universally). We may write a group binding as a sequence of individual bindings \( \langle a_1, s_1 \rangle \ldots \langle a_n, s_n \rangle \), where symbols \( \langle \rangle \) and \( [] \) are split and used as delimiters.

We use abbreviations \( T = p \lor \lnot p, \bot = \lnot T \) and \( \varphi \rightarrow \varphi' = \lnot \varphi \lor \varphi' \) for boolean connectives, \( F_\varphi = TU \varphi, G_\varphi = \lnot F \varphi \) for temporal operators, and finally \( \forall s, \varphi = \lnot \exists s, \neg \varphi \).

A variable \( s \) appears free in a formula \( \varphi \) if it appears out of the scope of a strategy quantifier, and an agent \( a \) appears free in \( \varphi \) if a temporal operator (either \( X \) or \( U \)) appears in \( \varphi \) out of the scope of any binding (in which case all agents are free, since group bindings treat all agents at once). We let \( free(\varphi) \) be the set of variables and players that appear free in \( \varphi \). If \( free(\varphi) \) is empty, \( \varphi \) is a sentence.

2.2 Models

We present concurrent game structures with imperfect information on both positions and actions. Towards our definition of the informed semantics, we specify for each agent \( a \) a set of agents \( A_0 \) whose strategies agent \( a \) is informed of.

Definition 2. A concurrent game structure (CGS) is a tuple \( G = \langle Ac, V, E, \ell, V_i, (\sim_a)_{a \in Ag}, (A_0 a)_{a \in Ag} \rangle \) where \( Ac \) is a finite set of actions, \( V \) is a finite set of positions, \( E : V \times Ac^g \rightarrow V \) is a transition function, \( \ell : V \rightarrow 2^AP \) is a valuation function, \( V_i \subseteq V \) is a set of initial positions, \( \sim_a \subseteq (V \times V) \cup (Ac \times Ac) \) is an equivalence relation called an observation relation, and \( A_0 a \subseteq Ag \) is the set of agents whose strategies agent \( a \) is informed of.

As in [Puchala, 2010] we consider equivalence relations on positions and actions. This, together with the inclusion of actions in histories (see below), allows us to model public actions in a very natural way.

Joint actions. In a position \( v \in V \), each player \( a \) chooses an action \( \alpha^a \in Ac \) and the game proceeds to position \( E(v, \alpha) \), where \( \alpha \in Ac^g \) stands for the joint action \( (\alpha^a)_{a \in Ag} \). Given a joint action \( \alpha^a = (\alpha^a)_{a \in Ag} \) and \( a \in Ag \), we let \( \alpha^a = \alpha^a \).

Histories, plays and strategies. A history (resp. play) is a finite (resp. infinite) sequence of positions and joint actions \( p = v_0\alpha_1 v_1 \ldots a_n v_n \) (resp. \( \pi = v_0\alpha_1 v_1 \ldots \)) such that \( v_0 \in V \) and for all \( i \) with \( 0 \leq i < n \) (resp. \( i \geq 0 \)), \( E(v_i, \alpha_{i+1}) = v_{i+1} \). For a history \( \rho = v_0\alpha_1 v_1 \ldots a_n v_n \) we let last(\( \rho \)) = \( v_n \) and \( |\rho| = n \). For a play \( \pi = v_0\alpha_1 v_1 \ldots \) and \( i \geq 0 \), we write \( \pi_{<i} = v_0 \ldots \alpha_i v_i \) for its prefix history ending at position \( i \).

We define \( \rho_{\leq i} \), similarly for histories \( \rho \) when \( i \leq |\rho| \). We let Hist be the set of histories. A strategy is a function \( \sigma : Hist \rightarrow Ac \), and we let Str be the set of all strategies.

Assignments. An assignment \( \chi : Ag \cup Var \rightarrow Str \) is a partial function assigning strategies to agents and variables. Given a tuple of strategies \( \sigma \in Str^Ag \), \( \chi[Ag \rightarrow \sigma] \) is the assignment that maps each agent \( a \) to \( \sigma^a \), and is equal to \( \chi \) on variables; similarly, \( \chi[s \mapsto \sigma] \) is the assignment that maps variable \( s \) to \( \sigma \) and is otherwise equal to \( \chi \). When \( \chi \) is defined for all agents and variables we call it total. Given a total assignment \( \chi \) and a set of agents \( A \subseteq Ag \), we let \( \chi \mid A \) be the restriction of \( \chi \) to \( A \cup Var \). A history \( \rho = v_0\alpha_1 \ldots a_n v_n \) is consistent with \( \chi \) if, for every \( i < n \) and agent \( a \) in the domain of \( \chi^a_{i+1} = \chi(a)(\rho_{\leq i}) \).

Outcomes. For a total agent assignment \( \chi \) and a history \( \rho \), we let Out(\( \chi, \rho \)) be the only play \( r \) that extends \( \rho \) consistently with \( \chi : \pi = \rho \alpha_1 v_1 \alpha_2 v_2 \ldots \) where for all \( i \geq 0 \) and \( a \in Ag \), \( \chi^a_{i+1} = \chi(a)(\pi_{\leq |\rho|+i}) \) and \( v_{i+1} = E(v_i, \alpha_{i+1}) \).

Indistinguishability. Two positions \( v, v' \) are indistinguishable to agent \( a \) if \( v \sim_a v' \), and similarly for actions. Two joint actions \( \alpha \) and \( \beta \) are indistinguishable to agent \( a \), written \( \alpha \sim_a \beta \), if they are component-wise indistinguishable, i.e., if \( \alpha^b \sim \beta^b \) for each \( b \in Ag \).

Synchronous perfect recall. We consider agents that remember their observations of all past positions and actions. Each observation relation \( \sim_a \) is thus extended to histories as follows: given two histories \( \rho = v_0\alpha_1 \ldots a_n v_n \) and \( \rho' = v'_0\alpha'_1 \ldots a_m v'_m \) we write \( \rho \sim_a \rho' \) if \( m = n \), \( v_i \sim_a v'_i \) for all \( i \in \{0, \ldots, n\} \), and \( \alpha_i \sim_a \alpha'_i \) for all \( i \in \{1, \ldots, n\} \).

2.3 Informed Semantics

We now define the informed semantics.
Informed perfect recall. To model agents that know some of the agents’ strategies, we refine the observation relation on histories as follows. Let $\chi$ be a total assignment, $a$ an agent and $A_a$ the subset of agents whose strategies $a$ knows. Two histories $\rho$ and $\rho'$ are indistinguishable for agent $a$, written $\rho \sim a \rho'$, if the following two conditions hold:

1. $\rho \sim a \rho'$
2. $\rho$ and $\rho'$ are both consistent with $\chi |_{A_a}$.

Given an agent $a$ and total agent assignment $\chi$, we say that a strategy $\sigma$ is $\sim a$-uniform if for all histories $\rho$ and $\rho'$ such that $\rho \sim a \rho'$ it holds that $\sigma(\rho) = \sigma(\rho')$.

**Remark 1.** Relation $\sim a$ is not an equivalence relation (it is not reflexive): histories inconsistent with $\chi |_{A_a}$ are not related to any history; it is an equivalence relation plus isolated points. The only impact this has on the semantics concerns nested knowledge operators, on which it has interesting consequences that we do not discuss here for lack of space.

Indistinguishable assignments. When evaluating an epistemic formula $K_a \varphi$ in a history $\rho$ and assignment $\chi$, existing variants of SL with knowledge such as [Maubert and Murano, 2018] and [Belardinelli et al., 2017a] evaluate $\varphi$ in all equivalent histories $\rho' \sim a \rho$ with the same assignment $\chi$. While quantifying on equivalent histories for the usual (uninformed) relation $\sim$, models agents that do not know the strategies being used, fixing the assignment $\chi$ indicates on the other hand agents that know all agents’ strategies. As a consequence of this clash, this semantics neither properly captures the knowledge of agents who are unaware of each other’s strategies, nor does it properly model the knowledge of agents who do know each other’s strategies, as we show in the next section.

To remedy this we introduce an equivalence relation on assignments. Two assignments $\chi$ and $\chi'$ are indistinguishable to agent $a$, written $\chi \sim a \chi'$, if they agree on all variables, and on the agents’ strategies that $a$ is informed of. Formally, $\chi \sim a \chi'$ if $\chi' |_{A_a} = \chi |_{A_a}$. The semantics of knowledge will quantify over all equivalent histories and assignments.

Interpreting strategy quantification. In introducing imperfect information to Strategy Logic, it is difficult to determine for whom a strategy should be uniform, at the level of strategy quantification. One solution is to specify it explicitly, as in [Berthom et al., 2021; Maubert and Murano, 2018]. Another is to require that it be uniform for all agents that are bound to it after the quantification, as in [Belardinelli et al., 2020]. However neither of these solutions can be used to define the informed semantics. Indeed, the informed perfect-recall relation $\sim a$ depends on the assignment $\chi$, which is only known after all agents have been bound to their strategy.

We thus propose a new way of interpreting the uniformity of strategies: instead of putting the uniformity constraint at the level of strategy quantification, we postpone it until after the group binding. In addition to allowing for a natural definition of the informed semantics, it is also a neat solution to defining the semantics of SL with imperfect information, independent of the matter of informed knowledge.

There are two dual ways of dealing with the uniformity constraint. Either we require that the strategy being assigned should be uniform and the remaining subformula should hold, or we say that if the strategy being assigned is uniform, then the subformula should hold. The first case corresponds to existentially quantified strategies, the second to universally quantified ones. Given a group binding $(Ag, s)$, we let $Ag^0 = \{ a \in Ag | s(a) = (s, i) \text{ for some } s \}$ and $Ag^1 = \{ a \in Ag | s(a) = (s, i) \text{ for some } s \}$.

**Definition 3.** Let $G = \{ Ac, V, E, \ell, V_c, \{ - \} \in Ag, \{ A_0 \} \in Ag \}$ be a CGS, $\chi$ a total assignment, $\rho$ a history and $\varphi$ an SLK_uni[BG]-formula. We define $G, \chi, \rho \models \varphi$ as follows:

- $G, \chi, \rho \models p$ if $p \in E(last(\rho))$
- $G, \chi, \rho \models \neg \varphi$ if $G, \chi, \rho \not \models \varphi$
- $G, \chi, \rho \models \varphi \lor \varphi'$ if $G, \chi, \rho \models \varphi$ or $G, \chi, \rho \models \varphi'$
- $G, \chi, \rho \models \exists s. \varphi$ if $\exists s \in Str s.t. G, \chi[s \rightarrow \sigma], \rho \models \varphi$
- $G, \chi, \rho \models (Ag, s) \varphi$ if $\forall a \in Ag^0 \chi'(a) \models \sim a \varphi' \text{ uniform, then }$ $\forall a \in Ag^1 \chi'(a) \models \sim a \varphi' \text{ uniform, }$
- $G, \chi', \rho \models \varphi$, where $\chi' = \chi[Ag \rightarrow s]$
- $G, \chi, \rho \models K_a \varphi$ if $\forall \rho' \in Hist s.t. \rho \sim a \rho'$ and $\forall \rho' s.t. \chi \sim a \chi'$, $v_i \chi', b, \rho' \models \varphi$
- and, writing $\pi = Out(\chi, \rho)$:
- $G, \chi, \rho \models X \varphi$ if $G, \chi, \rho \models \varphi$
- $G, \chi, \rho \models \varphi U \varphi'$ if $\exists i \geq 0 \text{ s.t. } G, \chi, s, \pi \in [\rho |+i] \models \varphi'$
- and $\forall j < i$, $G, \chi, s, \pi \in [\rho |+j] \models \varphi$

We explain the case for the binding operator, which is the novelty of this semantics. The idea is that the two dual bindings $\langle \rangle$ and $[]$ are used so that the first line restricts universal quantifications to uniform strategies by ignoring those that are not, while the second line forces existential quantifiers to pick uniform strategies. In both cases, uniformity of strategies is evaluated with respect to informed perfect recall.

The semantics of a sentence $\varphi$ does not depend on $\chi$, and we write $G, v_i \models \varphi$ if $G, \chi, v_i \models \varphi$ for some $\chi$.

**Public actions.** A CGS has public actions if for all $a \in Ac$ and $\rho \in Ag, \sim a \cap Ac \times Ac$ is the identity relation, meaning that all agents observe all actions perfectly.

**Hierarchical information.** A CGS has hierarchical information if there exists an enumeration $Ag = \{ a_1, \ldots, a_n \}$ of the agents such that $\sim a_{i+1} \subseteq \sim a_i$ for all $1 < i < n$.

**Remark 2.** The reason why we base our logic on the flat fragment of SL with Boolean Goals is already very expressive and can express, for instance, distributed synthesis, rational synthesis, and existence of Nash equilibria.
3 Illustrating the Informed Semantics

We discuss some aspects of our semantics, that we illustrate in the classic scenario of matching pennies.

\[ \text{Figure 1: CGS } \mathcal{G} \text{ for the game of matching pennies} \]

Consider the game of matching pennies \( \mathcal{G} \) illustrated in Figure 1, where the players are called 1 and 2 (loops on final positions are omitted). Each player has a coin that only they can see. First, player one places her hidden coin heads or tails up, then player 2 does the same with her hidden coin. The proposition \( m \) stands for the coins matching. We suppose that the players only observe their own actions, so \( v_3 \sim v_4, v_5 \sim v_6, v_1 \sim v_2, v_3 \sim v_5, \) and \( v_4 \sim v_6 \). In this example we omit the action names in the histories without ambiguity.

3.1 Impact on Knowledge

We first illustrate how restricting the quantification on equivalent plays to those that are consistent with known strategies affects the knowledge of the agents. The usual semantics for the knowledge operator in epistemic extensions of SL, that we can find for instance in [Maubert and Murano, 2018] and [Belardinelli et al., 2020], and that we shall refer to as “uninformed semantics” and denote \( =_u \) is the following:

\[ \mathcal{G}, \chi, \rho =_u K_a \varphi \text{ if for all } \rho', s.t. \rho' \sim \rho, \mathcal{G}, \chi, \rho' =_u \varphi \]

It differs from our semantics in two respects: (i) it uses the usual “uninformed” equivalence relation \( \sim \) on histories instead of the informed one \( \sim_2 \), and (ii) it does not quantify over indistinguishable assignments.

To illustrate the first difference (i), consider the formula

\[ \alpha = \forall s_1, \forall s_2, [1, s_1][2, s_2]XX (K_1m \lor K_1 \neg m) \]

which means that no matter which strategies are used, in the end player 1 knows whether the coins match or not. With the uninformed semantics it clearly does not hold, as player 1 does not observe player 2’s coin (so that \( v_0v_1v_3 \sim v_0v_1v_4 \) and \( v_0v_2v_5 \sim v_0v_2v_6 \)), nor does she know her strategy, so \( \mathcal{G}, v_0 \not= \alpha \). Now with the informed semantics, it is enough to assume that player 1 knows player 2’s strategy (i.e. \( 2 \in \mathcal{A}_1 \)) to make the formula true. Indeed, for any strategies \( s_1 \) and \( s_2 \) for players 1 and 2, the only history that player 1 considers possible with the uninformed relation \( \sim \) besides the actual outcome, is not consistent with strategy \( s_2 \) and is thus discarded by \( \sim_2 \). For instance if the outcome is \( v_0v_1v_3 \), i.e. \( s_1 \) and \( s_2 \) are the strategies that play heads, then the only history equivalent for \( \sim_2 \) \( v_0v_1v_4 \), in which player 2 plays tails, which is not consistent with \( s_2 \). So \( \mathcal{G}, v_0 \not= \alpha \).

We now illustrate the second difference (ii). Since the assignment does not impact the truth value of a formula without temporal operators, the only case in which considering equivalent assignments makes a difference is for formulas that involve knowledge about the future. So consider formula

\[ \beta = \forall s_1, \forall s_2, [1, s_1][2, s_2] (K_1XXm \lor K_1XX \neg m) \]

which means that for any pair of strategies \( s_1 \) and \( s_2 \), player 1 knows, before any move is made, whether in the end the coins will match. Intuitively this can only be the case if player 1 knows the strategies of both players. We check that indeed, with our semantics, the formula holds if we assume that \( \mathcal{A}_1 = \{1, 2\} \), and it does not if \( \mathcal{A}_1 \not= \{1, 2\} \). Fix a pair of strategies \( s_1, s_2 \), let \( \chi \) assign \( s_1 \) to 1 and \( s_2 \) to 2, and assume for instance that these strategies lead to matching coins, i.e. \( \mathcal{G}, \chi, v_0 = X X m \). First, note that the only history indistinguishable from \( v_0 \) is \( v_0 \). Now if \( \mathcal{A}_1 \not= \{1, 2\} \), meaning that player 1 does not know one of the strategies, then there exists at least one assignment \( \chi' \), \( \sim_2 \) \( \chi \) that has a different outcome, i.e. such that \( \mathcal{G}, \chi', v_0 = X X \neg m \). The same reasoning can be made if \( (s_1, s_2) \) lead to coins that do not match. As a result, \( \mathcal{G}, v_0 \not= \beta \). But now assume that player 1 knows both strategies, i.e., \( \mathcal{A}_1 = \{1, 2\} \). For any assignment \( \chi, \chi' \sim \chi \) implies that \( \chi' = \chi \). It follows that either \( \mathcal{G}, \chi, v_0 = K_1XXm \), or \( \mathcal{G}, \chi, v_0 = K_1XX \neg m \), or \( \mathcal{G}, \chi, v_0 = K_1XX \neg m \), or \( \mathcal{G}, v_0 = X X m \). So \( \mathcal{G}, v_0 = \beta \).

One can see that also with the usual “uninformed” semantics, it holds that \( \mathcal{G}, v_0 = \beta \) indeed, since it only considers the actual assignment, it corresponds in the previous paragraph to the case where \( \mathcal{A}_1 = \{1, 2\} \), i.e., the case where 1 knows both strategies. This is in contrast with what we illustrated with point (i), where the uninformed semantics corresponds to agents that do not know each other’s strategies, and it demonstrates an inconsistency in the usual semantics. The semantics we propose solves this problem.

3.2 Impact on Strategizing Power

Our semantics also makes a difference for agents’ strategic ability. Consider the following alternate “uninformed” semantics, which uses relation \( \sim_2 \) instead of \( \sim_1 \):

\[ \mathcal{G}, \chi, \rho =_u (Ag, \bar{s}) \varphi \text{ if } \]
\[ \forall a \in Ag_1, \chi'(a) \sim \chi_0 \text{ uniform, then } \]
\[ \forall a \in Ag_0, \chi'(a) \sim \chi_0 \text{ uniform, and } \]
\[ \nu, \chi', \rho = \varphi \text{, where } \chi' = \chi^2 \]

Consider again the example from Figure 1, and assume that player 2 knows player 1’s strategy, i.e., \( 1 \in \mathcal{A}_2 \). Formula

\[ \gamma = \exists s_2, \forall s_1, [1, s_1][2, s_2]XXm \]

means that player 2 has a strategy to ensure that the coins match. Knowing player 1’s strategy, intuitively this should be true: if player 1 plans to play heads, then play heads, otherwise play tails. However with the semantics that uses the usual “uninformed” notion of uniformity, this formula does not hold: indeed, because \( v_1 \) and \( v_2 \) are indistinguishable to
player 2, any \( \sim_2 \)-uniform strategy \( s_2 \) must play the same action in histories \( v_0 t v_1 \) and \( v_0 t v_2 \). If it is heads, then the coins will not match if 1 plays tails, and vice versa. So \( \mathcal{G}, v_0 \models \gamma \).

However with our informed semantics, the formula holds. Indeed consider the strategy \( s_2 \) such that \( s_2(v_0, v_1) = h \) and \( s_2(v_0, v_2) = t \). Clearly it ensures that the coins match. We just need to see that it is also uniform for the informed notion of uniformity. Let \( s_1 \) be a strategy and \( \chi = [1 \rightarrow s_1, 2 \rightarrow s_2] \). To see that \( s_2 \) is \( \sim_2 \)-uniform, observe that either \( v_0 t v_1 \) or \( v_0 t v_2 \) is not consistent with \( s_1 \). Therefore it is not the case that \( v_0 t v_1 \sim_2 v_0 t v_2 \), and \( s_1 \) does not have to assign the same action in both histories. As a result, \( \mathcal{G}, v_0 \models \gamma. \)

4 Model Checking SLK\textsubscript{inf}[BG]

We consider the model-checking problem for SLK\textsubscript{inf}[BG] which consists in deciding, given a CGS \( \mathcal{G} \), an initial position \( v_i \in \mathcal{V} \), and an SLK\textsubscript{inf}[BG] sentence \( \Phi \), whether \( \mathcal{G}, v_i \models \Phi \).

4.1 Undecidability for Hierarchical Information

It is proved in [van der Meyden and Wilke, 2005] that distributed synthesis for LTL with knowledge is undecidable, even for systems with hierarchical information. The setting there corresponds to players that know the strategy of everyone but the environment, which we can model in our informed semantics. Since distributed synthesis for LTLK objectives is expressible in SLK\textsubscript{inf}[BG], we obtain that:

Theorem 1. Model checking SLK is undecidable, even when restricted to CGS with hierarchical information.

4.2 Decidability for Public Actions

We now show that model checking SLK\textsubscript{inf}[BG] is decidable when all actions are public. This is in line with the results in [van der Meyden and Wilke, 2005], where the informed semantics is used implicitly, and distributed synthesis for LTLK is shown to be decidable on broadcast systems, which are very close to our CGS with public actions.

As in [Laroussinie et al., 2015; Berthon et al., 2021; Belardinelli et al., 2020], we solve the model-checking problem by reducing it to the model-checking problem for QCTL*, for which we briefly recall the syntax and semantics.

Definition 4. The syntax of QCTL\textsuperscript{*} is defined as follows:

\[
\varphi ::= p \mid \neg \varphi \mid \varphi \lor \varphi \mid E \psi \mid \exists p \varphi \\
\psi ::= \varphi \mid \neg \psi \mid \psi \lor \psi \mid X \psi \mid \psi U \psi
\]

where \( p \in \text{AP} \).

Note that the set \text{AP} may not be the same as in the previous section (in the reduction from SLK\textsubscript{inf} to QCTL\textsuperscript{*} we will use a bigger finite set of atomic propositions for QCTL\textsuperscript{*}).

We consider the tree semantics for QCTL\textsuperscript{*}, where formulas are evaluated on infinite trees, typically representing unfoldings of finite Kripke structures.

Trees. Let \( \Sigma \) be a finite alphabet. An infinite tree over \( \Sigma \) and \( \text{AP} \) is a mapping \( t : \Sigma^* \rightarrow 2^\text{AP} \). An element \( x \in \Sigma^* \) is a node, the empty word \( \epsilon \) is the root, a node \( y \) is a child of a node \( x \) if it is of the form \( y = x \cdot d \) with \( d \in \Sigma \), and \( t(x) \) is the label of node \( x \). A path \( \lambda = x_0 x_1 \ldots \) is an infinite sequence of nodes such that for all \( i \geq 0 \), \( x_{i+1} \) is a child of \( x_i \). We let \( \lambda_i = x_i \) denote the node at position \( i \), and \( \lambda_{>i} = x_i x_{i+1} \ldots \) is the suffix of \( \lambda \) starting at position \( i \). We say that \( \lambda \) starts in node \( x_0 \), and let \( \text{Paths}(x) \) be the set of paths that start in \( x \). The subtree of \( t \) rooted in node \( x \) is the tree \( t_x : y \mapsto t(x \cdot y) \). A tree is regular if it has finitely many different subtrees. Finally, given a tree \( t \) and an interpretation \( X_p \subseteq \Sigma^* \) for an atom \( p \in \text{AP} \), we define the tree

\[
t \otimes X_p : x \mapsto \begin{cases} (t(x) \cup \{ p \}) & \text{if } x \in X_p \\ (t(x) \setminus \{ p \}) & \text{otherwise} \end{cases}
\]

Definition 5. Let \( t \) be a tree, \( x \) a node and \( \lambda \) a path. The semantics of QCTL\textsuperscript{*} is defined as follows (we omit negation and disjunction):

\[
\begin{align*}
t, x &\models \psi & \text{if} & \psi \\
t, x &\models E \psi & \text{if} & \exists \lambda \in \text{Paths}(x) \text{ s.t. } \lambda \models \psi \\
t, x &\models \exists p \varphi & \text{if} & \exists X_p \subseteq \Sigma^* \text{ s.t. } t \otimes X_p, x \models \varphi \\
t, \lambda &\models \varphi & \text{if} & \lambda_0 \models \varphi \\
t, \lambda &\models X \psi & \text{if} & \lambda_{>0} \models \psi \\
t, \lambda &\models \psi U \psi' & \text{if} & \exists \lambda \geq 0 \text{ s.t. } \lambda_{>0} \models \psi' \text{ and } \forall \lambda' < \lambda, \lambda_{>\lambda'} \models \psi
\end{align*}
\]

We write \( t \models \varphi \) for \( t, \epsilon \models \varphi \).

For the rest of this section we fix a CGS \( \mathcal{G} = (\mathcal{A}, \mathcal{V}, V, E, t, V_i, \{ \sim_a \}_{a \in \mathcal{A}_g}, \{ A_a \}_{a \in \mathcal{A}_g}) \) with public actions and an SLK\textsubscript{inf}[BG] sentence \( \Phi \). The reduction is based on the one in [Belardinelli et al., 2020], where the two main ideas are that when actions are public and transitions deterministic, (1) each history is entirely determined by its initial position and the sequence of joint actions, and (2) two indistinguishable histories share the same sequence of joint actions. Note that only the first point relies on determinism.

We thus work on the tree consisting of all sequences of joint actions. A node \( \alpha = \alpha_1 \ldots \alpha_n \) in this tree represents the set of all histories \( p \) sharing this sequence of joint actions. By point 2 above, two histories that are indistinguishable for some agent are represented by the same node. This makes it possible to check uniformity of strategies locally in the tree.

Formally, let \( \Sigma = \mathcal{A}_g \) and for a node \( x \in \Sigma^* \) and an initial position \( v \in V_i \), we let \( \rho^v_x \) be the history in \( \mathcal{G} \) that starts in \( v \) and has sequence of joint actions \( x \). This is uniquely defined because the transition function is total and deterministic, and all histories can be written in this form. Point 2 above can be formalized as:

Lemma 2. If \( \rho^v_x \sim \rho^v_y \) then \( x = y \).

We introduce propositions \( \text{AP}_{\text{act}} = \{ p_{\alpha} \mid \alpha \in \mathcal{A}_g \} \) that mark in each node the last joint action performed, and propositions \( \text{AP}_{\text{pos}} = \{ p_v \mid v : V_i \rightarrow V \} \) that indicate the current position \( \nu(v) \) given that the initial position was \( v_i \). We then define the tree \( t_{\mathcal{G}} \) over \( \Sigma \) and \( \text{AP}_{\text{act}} \cup \text{AP}_{\text{pos}} \) as

\[
t_{\mathcal{G}} : \alpha_1 \ldots \alpha_n \mapsto \{ p_{\alpha_1}, p_v \}
\]

where \( \nu : v \rightarrow \text{last}(\alpha_1 \ldots \alpha_n) \). So each node is labeled by exactly one atom in \( \text{AP}_{\text{act}} \) and one in \( \text{AP}_{\text{pos}} \). Also, since \( \mathcal{G} \) is finite, the tree \( t_{\mathcal{G}} \) is regular.

To keep track of strategies we introduce propositions \( \text{AP}_{\text{str}} = \{ p_s^e \mid \tau : V_i \rightarrow \mathcal{A} \text{ and } s \in \text{Var}(\Phi) \} \), where \( \tau \) is
called a local strategy and \( \text{Var}(\Phi) \) is the set of strategy variables in \( \Phi \) (we assume without loss of generality that each variable is quantified at most once in \( \Phi \)). Proposition \( p^s_x \) in node \( x \) means that on history \( \rho^s_x \), strategy \( s \) chooses action \( \tau(v) \). To represent agents’ knowledge we introduce propositions \( \text{AP}_i = \{ p^a_x : a \in \text{Ag} \text{ and } i \subseteq V_i \times V_i \} \), where \( i \approx \) is called a local observation relation. The intended meaning of \( p^a_x \) holding in a node \( x \) is that histories \( \rho_x^a \) and \( \rho_x^c \) are indistinguishable to agent \( a \) if and only if \( v \approx u \). By Lemma 2, indistinguishability relations can be completely characterized by such propositions. To capture the interpretation of “indistinguishable to agent \( a’ \)” corresponding to the informed semantics, these propositions will be placed on the tree after those encoding strategies, and consistently with them.

We define for each subformula \( \varphi \) of \( \Phi \) a QCTL* formula \( [\varphi]_{V_i} \). The translation is parameterized by a mapping \( B : \text{Ag} \rightarrow \text{Var} \) that represents the current binding of agents to strategy variables, and a position \( v_i \in V_i \) representing the initial position of the current history. For atomic propositions:

\[
[p]_v^B = \bigvee_{\nu \in \text{Var}(\ell(v_i))} p_{\nu}
\]

This formula grabs the unique \( \nu \) for which \( p_{\nu} \) holds in the current node, and checks whether \( p \) holds in the current position as indicated by \( \nu \) given that the initial position is \( v_i \). Boolean connectives simply distribute:

\[
[-\varphi]_{v_i}^B = [-\varphi]_{v_i}^B \quad [\varphi \lor \varphi']_{v_i}^B = [\varphi]_{v_i}^B \lor [\varphi']_{v_i}^B
\]

For strategic quantification, let \( \{\tau_1, \ldots, \tau_k\} \) be an enumeration of the possible local strategies, with \( \mathbb{K} = \{\text{Ag}\}^{V_i} \). Let

\[
[\exists s. \varphi]_{v_i}^B = \exists p^s_{\tau_1} \ldots \exists p^s_{\tau_k}; \text{strat}_s \land [\varphi]_{v_i}^B
\]

where

\[
\text{strat}_s = \text{Ag} \bigvee_i (p^s_{\tau_i} \land \bigwedge_{j \neq i} \neg p^s_{\tau_j})
\]

checks that each node is labeled by exactly one local strategy.

Next comes the binding operator \( (\text{Ag}, \hat{s}) \), which we represent as a mapping \( B^s : \text{Ag} \rightarrow \text{Var} \times \{\ell, \{\}, \} \). Let \( \{a_1, \ldots, a_m\} \) be an enumeration of the agents and \( \{\approx_1, \ldots, \approx_l\} \) an enumeration of the possible equivalence relations over \( V_i \). For each agent \( a \) we define the initial local observation relation \( \approx_a = \approx \cap V_i \times V_i \). Also, given a local observation relation \( \approx_a \) for agent \( a \) (in some node \( x \)), a tuple \( \tau \) of local strategies for each agent, as well as a next joint action \( \alpha \) and resulting current positions \( \nu \) (in node \( x', \alpha \)), we define the updated local observation relation for agent \( a \) as \( \approx_a' = (\approx_a, \tau, \alpha, \nu) = \)

\[
\left\{ (v_i, u_i) \mid v_i \approx_a u_i, \mu(v_i) \approx_v \mu(u_i) \text{ and } \forall b \in \text{Ag}, \tau_b(v_i) = \alpha_b = \tau_b(u_i) \right\}
\]

This definition means that two histories \( \rho_{v, x}^{a, \alpha} \) and \( \rho_{u, x}^{a, \alpha} \) are indistinguishable for \( a \) if \( \rho_{v, x}^{a, \alpha} \) and \( \rho_{u, x}^{a, \alpha} \) are indistinguishable (\( v_i \approx_a u_i \)), the next positions \( \text{last}(\rho_{v, x}^{a, \alpha}) \) and \( \text{last}(\rho_{u, x}^{a, \alpha}) \) are indistinguishable (\( \mu(v_i) \approx_v \mu(u_i) \)), and for each agent \( b \) whose strategy is known to \( a \), the last action \( \alpha_b \), played by \( b \) corresponds to strategy \( \tau_b \) in both \( \rho_{v, x}^{a, \alpha} \) and \( \rho_{u, x}^{a, \alpha} \) (so \( \rho_{v, x}^{a, \alpha} \) and \( \rho_{u, x}^{a, \alpha} \) are consistent with \( \tau_b \)). Given a binding \( B \), we let \( \text{Ag} \) (resp. \( \text{Ag} \)) be the set of agents \( a \) such that \( B(a) = (s, (\{\})) \) (resp. \( (s, []) \)) for some variable \( s \). Define

\[
[B^s]_{v_i}^B = \exists p^a_{\tau_1} \ldots \exists p^a_{\tau_m}; \text{strat}_s \land [\varphi]_{v_i}^B
\]

where \( \text{uniq}_a = \text{Ag} \bigvee(i \approx_a \land \bigwedge_{b \in \text{Ag}} p^b_{\tau_b}) \) checks each node is marked with one unique local observation relation,

\[
\text{inf}_a = p^a_{\tau_a} \land (\text{Ag} \bigvee (p^a_{\tau_a} \land \bigwedge_{b \in \text{Ag}} p^b_{\tau_b}))) \rightarrow
\]

\[
\text{AX} \bigcap (p_{\alpha} \land p_{\nu}) \rightarrow p^a_{\tau_a} \approx_{a, \tau, \alpha, \nu})
\]

checks that the local observation relation is initially correct and that in each step it is updated correctly, and

\[
\text{loc-unif}(\approx, \tau) = \left\{ \begin{array}{ll}
\text{true} & \text{if for all } u_i, u'_i \in V_i \text{ s.t. } u_i \approx u'_i,
\text{false} & \text{otherwise}.
\end{array} \right.
\]

For the knowledge operator, let

\[
[K_a \varphi]_{v_i}^B = \bigwedge_{a \approx_{V_i}} (p^a_{\tau_a} \land \bigwedge_{b \in \text{Ag}} [\varphi]_{v_i}^B)
\]

We are left with the temporal operators:

\[
[X \varphi]_{v_i}^B = \bigwedge_{a \approx_{V_i}} (p^a_{\tau_a} \land \text{AX} \left( p_{\alpha} \land p_{\nu} \rightarrow [\varphi]_{v_i}^B \right)
\]

where \( \alpha(\tau, v_i) \) is the joint action where each agent \( a \) plays action \( \tau_a(v_i) \). Similarly, we let

\[
A \left( \bigwedge_{a \approx_{V_i}} (p^a_{\tau_a} \land \text{AX} \left( p_{\alpha} \land p_{\nu} \rightarrow [\varphi]_{v_i}^B \right) \bigg) \right) =
\]

\[
\text{Lemma 3. For every initial position } v_i \in V_i \text{ and every initial binding } B, \text{ it holds that } G, v_i \models \Phi \iff G, v_i \models [\Phi]_{v_i}^B.
\]

Because model checking QCTL* on regular trees is decidable [Laroussinie and Markey, 2014], we obtain that

\[
\text{Theorem 4. Model checking SLK}_{\text{int}} \text{ on CGS with public actions is decidable.}
\]

5 Conclusion and Future Work

In this work we defined the first semantics for Strategy Logic that can model agents that know other agents’ strategies. This semantics reflects the intuitive idea that knowing others’ strategies refines agents’ knowledge about the system, and increases their ability to strategize. It is also general enough to capture both situations found in the literature, which correspond to the two extreme cases. Two important questions left for future work are the implications of the informed semantics on higher-order knowledge, and trying to extend this new informed semantics to the full syntax of \( \overline{\text{SL}} \).
References


