Verification of Broadcasting Multi-Agent Systems against an Epistemic Strategy Logic

Francesco Belardinelli
Laboratoire IBISC, UEVE and IRIT Toulouse
belardinelli@ibisc.fr

Alessio Lomuscio
Department of Computing
Imperial College London
a.lomuscio@imperial.ac.uk

Aniello Murano and Sasha Rubin
DIETI
Università degli Studi di Napoli
murano@na.infn.it
rubin@unina.it

Abstract

We study a class of synchronous, perfect-recall multi-agent systems with imperfect information and broadcasting, i.e., fully observable actions. We define an epistemic extension of strategy logic with incomplete information and the assumption of uniform and coherent strategies. In this setting, we prove that the model checking problem, and thus rational synthesis, is non-elementary decidable. We exemplify the applicability of the framework on a rational secret-sharing scenario.

1 Introduction

Epistemic logic has a long tradition in knowledge representation and reasoning, multi-agent systems (MAS), and more broadly in artificial intelligence [Meyer and Hoek, 1995]. A significant line of research over the past twenty years has concerned its combination with various temporal logics such as LTL, CTL, and the like [Clarke et al., 2002]. The resulting syntax can express a wide range of properties of multi-agent systems, including the knowledge agents have about the world, about each other’s knowledge, how this evolves over time and whether sophisticated epistemic states such as common knowledge are acquired in a system’s run [Halpern and Vardi, 1989].

Temporal-epistemic properties of multi-agent systems have been studied under a variety of assumptions, including synchronicity, asynchronicity, perfect recall, bounded recall, no learning, and observational semantics [Fagin et al., 1995]. These aspects are now known to impact the resulting axiomatisations [Halpern et al., 2003; Belardinelli and Lomuscio, 2009] as well as the complexity of the verification problem [Meyden and Shilov, 1999]. For these reasons, a key aspect in this line of work has been the identification of expressive fragments with relatively low complexity.

Recently there has been considerable interest in the extension of the formalisms above to languages sufficiently expressive to capture strategic abilities of agents. Towards this aim, alternating-time temporal logic (ATL) [Alur et al., 2002] and strategy logic (SL) [Mogavero et al., 2014] have been put forward and combined with epistemic modalities and uniform strategies [Hoek and Wooldridge, 2003; Belardinelli, 2014; Huang and Meyden, 2014; Čermák et al., 2014; Berthon et al., 2017a].

Reasoning about strategic abilities of MAS under imperfect information is known to be difficult. For example, model checking MAS against ATL specifications under incomplete information goes from PTIME-complete to \( \Delta^2_p \)-complete under memoryless strategies (i.e., imperfect recall) [Jamroga and Dix, 2006] and is undecidable under memoryfull strategies (i.e., perfect recall) [Dima and Tiplea, 2011]. For this reason it is of interest to identify expressive classes of MAS for which the model checking problem is decidable. The aim of this paper is to make a contribution in this direction.

Specifically, we introduce ESL, an epistemic extension of SL based on synchronous perfect-recall strategies (Section 2). The language introduced can express rational synthesis [Fisman et al., 2010; Wooldridge et al., 2016; Kupferman et al., 2016], but its model-checking problem is undecidable. However, we identify a significant class BA-iCGS of systems: those having broadcast (i.e., fully observable) actions (Section 2.2) and prove that model checking BA-iCGS against ESL is non-elementary decidable (Section 4). This is a tight result as a matching lower-bound already holds in the perfect-information case. We illustrate our formalism on a rational secret-sharing scenario with broadcast actions (Section 3).

Related Work. As mentioned above, several approaches have been put forward to reason about strategies and knowledge in the context of MAS.

SL and knowledge have been combined before in the context of MAS. In [Cermák, 2014; Čermák et al., 2014], an epistemic variant of SL [Mogavero et al., 2014] was introduced. However, this was limited to epistemic sentences, whereas we consider the full combined language, and the approach assumed observational semantics, whereas we here consider synchronous perfect recall. Although not studied in these papers these formalisms have an undecidable model checking problem if evaluated under synchronous perfect recall. Also, [Berthon et al., 2017b] defines a variant of SL with uniform strategies. They achieve decidability by a variation of the tradition of assuming a hierarchy on the observations. In this paper we do not make any hierarchical assumptions.

A key aspect of the work here presented is that it relies on broadcasting to achieve decidability in the context of a very expressive specification language. The notion of broadcast has already been studied in the context of knowledge [Fa-
gin et al., 1995; Lomuscio et al., 2000]. A further important result in this area is that for broadcast systems the synthesis problem of specifications in LTL and knowledge is decidable [Meyden and Wilke, 2005]. However, ESL is strictly more expressive and synthesis, which in our case can be expressed via model checking, can also be shown to be decidable. An approach to reasoning about strategies and knowledge under broadcast was also recently presented in [Belardinelli et al., 2017]. However, their logic is considerably less expressive than ours, as it is based on ATL and not SL. In particular, it cannot express Nash equilibria and rational synthesis, which are essential features of this contribution.

Rational synthesis has been studied before in the context of perfect information. In [Kupferman et al., 2016] the strong-rational synthesis problem with LTL objectives (and aggregation of finitely many objectives), is shown to be 2EXPTIME-complete. In [Gutierrez et al., 2017], Equilibrium Logic is introduced to reason about Nash equilibria in games with LTL and CTL objectives. However, both cases assume perfect information of the agents. Synthesis under imperfect information has been first tackled in [Gutierrez et al., 2016] albeit for a restricted class of CGS, viz. reactive modules. In this paper we explore synthesis in CGS under imperfect information.

2 Strategy Logic with Imperfect Information

In this section we present strategy logic (SL) (see [Mogavero et al., 2014] for a definition of SL) in an imperfect information setting. In particular, we introduce the class of imperfect information concurrent game structures (iCGS) with broadcast actions only (BA-iCGS). We start with some preliminaries. For an infinite or non-empty finite sequence $u \in X^\omega \cup X^+$ of elements in $X$, we write $u_i$ for the $(i+1)$-th element of $u$, i.e., $u = u_0 u_1 \ldots$. For $i \geq 0$, $u \leq_i$ is the prefix of $u$ of length $i+1$, i.e., $u \leq_i = u_0 u_1 \ldots u_i$. The empty sequence is denoted as $\epsilon$. The length of a finite sequence $u \in X^+$ is denoted as $|u|$. For a vector $v \in \prod_i X_i$, we denote the $i$-th co-ordinate of $v$ by $v(i)$. In particular, for $F \in \prod_i (X_i)^V$ we may write $F(i) \in X_i^V$ and $F(i)(y) \in X_i$.

2.1 iCGS

Hereafter we consider concurrent game structures enriched with indistinguishability relations. These are the standard setting for agent-based logics under imperfect information [Jamroga and van der Hoek, 2004; Bulling and Jamroga, 2014].

Definition 1 (iCGS). An imperfect information concurrent game structure (iCGS) is a tuple $S = \{Ag, AP; \{Act_a\}_{a \in Ag}, S, S_0, tr, (\sim_a)_{a \in Ag}, \lambda\}$, where:

1. Ag is the finite non-empty set of agent names.
2. AP is the finite non-empty set of atomic propositions.
3. Act$_a$ is the finite non-empty set of actions for $a \in Ag$; for $A \subseteq Ag$, let Act$_A = \cup_{a \in A} Act_a$, and let Act = Act$_{Ag}$.
4. S is the finite non-empty set of states and $S_0 \subseteq S$ is the non-empty set of initial states.
5. tr : $S \times ACT \rightarrow S$ is the transition function, where ACT = $\prod_{a \in Ag} Act_a$ is the set of all joint actions.
6. $\sim_a \subseteq S^2$ is the indistinguishability relation for agent $a$, which is an equivalence relation.

7. $\lambda : AP \rightarrow 2^S$ is the labelling function that assigns to each atom $p$ the set of states $\lambda(p)$ in which $p$ holds.

A concurrent game structure (CGS) is an iCGS for which $\sim_a = \{(s, s) : s \in S\}$ for all $a \in Ag$. This corresponds to the perfect-information setting [Alur et al., 2002].

We now define what it means for an agent to have synchronous perfect-recall in an iCGS $S$. A history in $S$ is a non-empty finite sequence $h_0 h_1 \ldots$ in $S^+$ such that for all $i \geq 0$, there exists a joint action $J_i \in Act$ such that $h_{i+1} \in tr(h_i, J_i)$. The set of all histories in $S$ is denoted as hist($S$), and the set of histories $h'$ that extend history $h$ is denoted as hist($S, h$), that is, $h \leq_i |h'| = h$.

Hereafter we use the following notation: if $\sim$ is a binary relation on $S$, we define the extension of $\sim$ to histories as the binary relation $\equiv$ on hist($S$) such that $h \equiv h'$ if $|h| = |h'|$ (i.e., synchronicity) and $h_j \sim h'_j$ for all $0 \leq j \leq |h|$ (i.e., perfect recall). We consider three instantiations for individual, common, and distributed knowledge respectively. If $\sim_a$ is the indistinguishability relation for agent $a$, then two histories $h, h'$ are indistinguishable to agent $a$, if $a$ is $h \equiv_a h'$. For $A \subseteq Ag$, let $\sim^A = \{(a \in A \sim_a)^*\}$, where $*$ denotes the reflexive and transitive closure (w.r.t. relation composition), and its extension to histories is denoted $\equiv^A$. For $A \subseteq Ag$, let $\sim_\ominus^A = \cap_a \sim_a \sim_a$, and its extension to histories is denoted $\equiv_\ominus^A$.

A deterministic memoryfull strategy, or simply strategy, is a function $\sigma : hist(S) \rightarrow Act$ (recall that Act = $\cup_{a \in Ag} Act_a$). The set of all strategies is denoted $\Sigma(S)$. Further, a strategy $\sigma_a$ is coherent for $a$ if for every $h \in hist(S)$, $\sigma_a(h) \in Act_a$; while $\sigma_a$ is uniform for $a$ if for all $h, h' \in hist(S)$, $h \equiv_a h'$ implies $\sigma_a(h) = \sigma_a(h')$. Then, a joint full strategy is a function $\sigma_{Ag} : Ag \rightarrow \Sigma(S)$ that associates to each agent $a \in Ag$ a strategy that is both coherent and uniform for $a$. We write $\sigma_{Ag}(a) = \sigma_a$. For every $s_0 \in S_0$, a joint full strategy $\sigma_{Ag}$ defines a unique infinite sequence $\pi(\sigma_{Ag}) = s_0 s_1 \ldots$ of states, i.e., for all $i \geq 0$, $s_{i+1} = tr(s_i, \sigma_{Ag}(s_0 s_1 \ldots s_i))$. A history $h$ is consistent with $\sigma_{Ag}$ if $h$ is a prefix of $\pi(\sigma_{Ag})$. Given $h \in hist(S)$, define the set out($h, \sigma_{Ag}$) of outcomes of $\sigma_{Ag}$ from $h$ as the set of histories $h' \in hist(S, h)$ that extend $h$ and are consistent with $\sigma_{Ag}$. Notice that for every $i \geq 0$, there is unique $h' \in out(h, \sigma_{Ag})$ of length $|h|+i$. Thus, write $\pi(h, \sigma_{Ag}) \in S^\omega$ for the infinite sequence all of whose prefixes are in out($h, \sigma_{Ag}$).

2.2 BA-iCGS—iCGS with Broadcast Actions only

In this paper we focus on a particular class of iCGS, those having broadcast actions only. This section is reported from [Belardinelli et al., 2017] (where these were called iCGS with public actions only).

Definition 2 (BA-iCGS). An iCGS $S$ only has broadcast actions if for every agent $a \in Ag$, states $s, s' \in S$, and joint actions $J, J' \in Act$, if $J \neq J'$ and $s \sim_a s'$ then $tr(s, J) \neq tr(s', J')$. In this case we call $S$ a broadcast iCGS. We write BA-iCGS for the set of broadcast iCGS.

Broadcast iCGS arise naturally in several MAS scenarios, including epistemic puzzles (e.g., the muddy children puzzle) and games (e.g., battleship). In Section 3 we discuss an application to rational synthesis.

We define the following natural encoding of histories.
Definition 3. Let $S$ be an iCGS. Define the encoding function $\mu : S_0 \times \text{ACT}^* \rightarrow \text{hist}(S)$ that maps $(s_0, u)$ to the history $h$ of length $|u| + 1$ and such that $h_0 = s_0$ and $h_j = \text{tr}(h_{j-1}, u_{j-1})$ for $1 \leq j \leq |u|$. 

In case $S$ is a BA-iCGS, then $\mu$ is a bijection, i.e., for every $h \in \text{hist}(S)$ there exists a unique $(s_0, u) \in S_0 \times \text{ACT}^*$ such that $\mu(s_0, u) = h$. Moreover, the moment different joint actions are taken, two histories become distinguishable:

Lemma 1. Let $S$ be a BA-iCGS. For all $a \in \mathcal{A}$, $u, u' \in \text{ACT}^*$ and $s, s' \in S_0$, if $\mu(s, u) \equiv_a \mu(s', u')$ then $u = u'$.

Proof. Indeed, if $\mu(s, u) \equiv_a \mu(s', u')$ then $|u| = |u'|$ and, for all $0 \leq j \leq |u|, \mu(s, u)_j \sim_a \mu(s', u')_j$. By the definition of having only broadcast actions, $u_j = u'_j$ for all $j < |u|$. $\square$

The next characterisation of uniformity in BA-iCGS follows from Lemma 1 and is central to our decidability result:

Proposition 1. Let $S$ be a BA-iCGS, and let $\sigma$ be a coherent strategy for agent $a$. Then $\sigma$ is uniform for agent $a$ if and only if for all $v \in \text{ACT}^*$, $s, s' \in S_0$ we have that $\mu(s, v) \equiv_a \mu(s', v)$ and uniform for $a$. An assignment $\sigma$ is $\varphi$-compatible if, for every $x \in \mathcal{V}$, the strategy $\chi(x)$ is coherent and uniform for every agent in $\text{shr}(x, \varphi)$. We define $(S, h, \chi) \models \varphi$ where $h \in \text{hist}(S)$, $\varphi$ is a formula, $\chi$ is a $\varphi$-compatible assignment, and $\pi := \pi(h, \chi)_a$ is the unique infinite sequence that extends $h$ by following the restriction of $\chi$ to $\mathcal{A}$:

$L_S(S, h, \chi) = p$ if $\text{last}(h) \in \chi(p)$, $p \in \mathcal{A}$.

$L_S(S, h, \chi) = \neg \chi_1$ if it is not the case that $(S, h, \chi) \models \varphi_1$.

$L_S(S, h, \chi) = \chi_1 \land \chi_2$ if $(S, h, \chi) \models \varphi_i$ for $i \in \{1, 2\}$.

$L_S(S, h, \chi) = \langle x \rangle \varphi_1$ if there exists a strategy $\sigma$ that is uniform and coherent for every agent in $\text{shr}(x, \varphi_1)$ such that $(S, h, \chi^x_\sigma) \models \varphi_1$.

$L_S(S, h, \chi) = \langle (x, a) \rangle \varphi_1$ if $(S, h, x^{\chi_1}_{\chi_2}) \models \varphi_1$.

$L_S(S, h, \chi) = \langle h \rangle \varphi_1$ if every history $h' \in \text{hist}(S)$, $h' \equiv_a h$ implies $(S, h', \chi) \models \varphi_1$.

$L_S(S, h, \chi) = \langle C \rangle \varphi_1$ if every history $h' \in \text{hist}(S)$, $h' \equiv_a h$ implies $(S, h', \chi) \models \varphi_1$.

$L_S(S, h, \chi) = \langle D \rangle \varphi_1$ if every history $h' \in \text{hist}(S)$, $h' \equiv_a h$ implies $(S, h', \chi) \models \varphi_1$.

$L_S(S, h, \chi) = X \varphi_1$ if $(S, \pi(x_{\nu+1,x}) \models \varphi_1$.

$L_S(S, h, \chi) = \varphi_1 \cup \varphi_2$ if there exists $i \geq |h|$ s.t. $(S, \pi(x_{\nu+1,x}) \models \varphi_2$ for all $j$ with $|h| \leq j < i$, $(S, \pi(x_{\nu+1,x}) \models \varphi_1$.

The following says that the satisfaction is well-defined:

Lemma 2. In the expressions $(S, h, \chi) \models \varphi$ on the right-hand sides, $\chi$ is always a $\varphi$-compatible assignment.

For any formula $\varphi$, we write $S \models \varphi$ to mean that $(S, s, \chi) \models \varphi$ for every $s \in S_0$ and assignment $\chi$ (observe that states are histories of length 1). One can prove (as usual) that the satisfaction of ESL-formulas depends only on their free variables and agents, that is, if assignments $\chi$ and $\chi'$ co-occur on free($\varphi$), then $(S, h, \chi) \models \varphi$ if and only if $(S, h, \chi') \models \varphi$. Thus, e.g., if $\varphi$ is a sentence, then $S, s \models \varphi$ if $(S, s, \chi) \models \varphi$ for some assignment $\chi$.

Clearly ESL extends SL. Indeed, one restricts the syntax of ESL to the epistemic-free fragment, and the models to CGS (i.e., $\equiv_a := \{(s, s) : s \in S\}$ for every $a \in \mathcal{A}$):

Proposition 2. For every SL sentence $\varphi$ there is an ESL sentence $\varphi$ s.t. for all iCGS $S$, we have that $S \models \varphi$ if $S \models \varphi$.

The proof simply requires to show that the state-based semantics of SL in [Mogavero et al., 2014] can be captured by our history-based semantics.

The next proposition says that ATL*K embeds in ESL (see [Jamroga and van der Hoek, 2004] for the definitions of ATL*K). Although the embedding is as expected, the proof that it is correct is subtle:

Proposition 3. For every ATL*K formula $\varphi$ there is an ESL sentence $\varphi$ s.t. for all iCGS $S$, we have that $S \models \varphi$ if $S \models \varphi$.

Proof. The main difficulty is to translate the ATL*K operator $\langle A \rangle$ in ESL, which we illustrate. Consider $\mathcal{A} = \{(a_1, a_2, a_3, a_4) : A \in \{1, 2\}\}$, and the ATL*K formula $\varphi = \langle A \rangle \psi$. The formula says that for every set of uniform strategies, one for each agent in $A$, every path consistent with these strategies satisfies $\psi$. Consider the ESL formula $\varphi = \langle (x_1), (x_2) \rangle [x_3] [x_4] (x_1, a_1) (x_2, a_2) (x_3, a_3) (x_4, a_4) \psi$. Clearly, $\varphi$ logically implies $\psi$ since the paths consistent with $x_1, x_2$ include those generated by uniform strategies $x_1, x_2, x_3, x_4$. On the
other hand, let $\pi$ be any path consistent with strategies $\sigma_1, \sigma_2$ (for agents $a_1, a_2$). It is sufficient to show that there exist uniform strategies, $\sigma_3$ for agent $a_3$ and $\sigma_4$ for agent $a_4$, such that $\pi(s, \sigma_{Ag}) = \pi$. Indeed, for every $n \geq 0$, let $J \in AC$ be a joint action such that i) $\sigma_3(\pi_{\leq n}) = J(i)$ for $i = 1, 2$, and ii) $tr(\pi_n, J) = \pi_{n+1}$. Define $\sigma_4(\pi_{\leq n}) = J(i)$ for $i = 3, 4$. Since the uniformity condition only restricts pairs of histories of the same length, we can extend $\sigma_3$ and $\sigma_4$ to uniform strategies. Note that $\pi(s, \sigma_{Ag}) = \pi$, as required.

We now introduce the main decision problem of this work.

**Definition 4** (Model Checking). Let $C$ be a class of iCGS and $F$ a sublanguage of ESL. Model checking $C$ against $F$ specifications is the following decision problem: given $S \in C$ and $\varphi \in F$ as input, decide whether $S \models \varphi$.

Model checking iCGS against ATL is undecidable [Dima and Tiplea, 2011]. Thus, applying Proposition 3, we get:

**Proposition 4.** Model checking iCGS against ESL is undecidable.

Indeed, it is undecidable even if $C$ consists of all iCGS with $|Ag| = 3$ and $F$ contains just the ATL formula $G p$, see [Dima and Tiplea, 2011]. The source of the undecidability is the interplay between two assumptions: a) $\sim_1$ and $\sim_2$ are incomparable under the refinement-order on equivalence relations, and b) agent 3 can privately communicate with agents 1 and 2. In the sequel we prove that model checking is decidable assuming all agents only have broadcast actions. Thus, we keep property a) while dropping property b).

### 3 Rational Synthesis under Imperfect Information

In this section we show how to express central game-theoretic properties in ESL, e.g., the existence of Nash equilibria in multi-player games of imperfect information with epistemic objectives. Moreover, we illustrate that ESL can be used to reason about rational secret-sharing, i.e., rational agents that communicate by broadcast actions in order to learn a secret whose “shares” have been distributed among them.

#### 3.1 Expressing Rational Synthesis in ESL

Several questions in computer science can be cast as the problem of deciding if there exists a joint winning strategy for a coalition of agents against a coalition of adversarial agents (and computing one if it exists). In the verification literature this problem is called synthesis.

However, as argued in [Wooldridge et al., 2016; Kupferman et al., 2016; Abraham et al., 2011], the partition of agents into “good” and “bad” is often insufficient, and it is more appropriate to view agents as rational. That is, agents have preferences over outcomes and act in a way that increases their own utility. Then, instead of reasoning about winning strategies, one should reason about rational strategy profiles, i.e., that satisfy some notion of equilibrium. Application domains include rational distributed computing and rational cryptography [Abraham et al., 2011], and negotiating systems with self-interested agents [Jennings and others, 2001]. Technically, suppose we are given an iCGS $S$ representing the multi-agent system, and ATLK-formulas $\gamma_a$ representing the objective of agent $a \in Ag$. Here, ATLK is the logic consisting of the set of path-formulas of ATL*K. We can then talk about Nash equilibria $\pi$ in games of the form $G = (S, \{\gamma_a\}_{a \in Ag})^1$. Rational synthesis considers the following decision problem (sometimes called E-NASH):

**Definition 5** (Rational Synthesis for ATLK objectives, cf. [Kupferman et al., 2016]). Given an iCGS $S$, ATLK-formulas $\gamma_a$ for every $a \in Ag$, and an ATLK-formula $\varphi$, decide whether there exists a Nash equilibrium $\pi$ in the game $G = (S, \{\gamma_a\}_{a \in Ag})$ such that the path induced by $\pi$ satisfies $\varphi$.

Intuitively, $\varphi$ represents some global property that the designer wants to ensure given that agents are self-interested. In case $\varphi = true$, this simply asks if there exists a Nash-equilibrium. Moreover, if there is such a Nash equilibrium, the synthesis problem concerns deriving one such strategy profile $\pi$. The dual problem, called Strong Rational Synthesis (sometimes called A-NASH), concerns deciding whether all Nash equilibria induce a path that satisfies $\varphi$ [Kupferman et al., 2016].

We now show that rational synthesis for ATLK objectives reduces to model checking against ESL. Suppose $Ag = \{a_1, a_2, \ldots, a_n\}$, and let $\pi$ be an $n$-tuple of variables. Let $\beta$ be the expression $(x_1, a_1)(x_2, a_2)\ldots(x_n, a_n)$ that binds agent $a_i$ to strategy $x_i$. The following formula $RatSyn_\pi(\pi)$ in ESL expresses that $\pi$ is a Nash equilibrium whose induced execution satisfies $\varphi$:

$$\varphi \land \bigwedge_{a \in Ag} (\langle y \rangle (y, a) \rightarrow \gamma_a)$$

In words, if agent $a_i$ uses $x_i$ then the resulting execution satisfies $\varphi$, and no agent has an incentive to unilaterally deviate from the strategy profile $\pi$. Then:

**Lemma 3.** Rational synthesis for ATLK objectives is reducible to model checking against the ESL-formula $\langle x_1 \rangle \ldots \langle x_n \rangle RatSyn_\pi(\pi)$.

A universally quantified formula is used for Strong Rational Synthesis. It is important to observe that ESL can express other equilibrium concepts such as subgame-perfect equilibria, concepts that capture deviations by groups of players such as $k$-resilience and $t$-immunity, and the combination $(k, t)$-robustness that captures fault-tolerance [Abraham et al., 2011]. Also, ESL is able to express the existence of Nash equilibria w.r.t. epistemic objectives, which, to the best of our knowledge, has not yet been considered in the literature. We illustrate this last point in the next section.

### 3.2 Rational Secret-Sharing with Broadcast

We illustrate the model-checking problem for BA-iCGS against ESL with a simple scenario inspired by [Abraham et al., 2006] that uses broadcast. In the classic $m$-out-of-$n$ secret-sharing problem, for $Ag = \{1, 2, \ldots, n\}$, initially each agent $i \in Ag$ privately holds a “share” $f_i$ of a secret $f_0$, and any $m$ “good” agents can collaborate to learn the secret

---

1The framework can also support every agent having finitely-many Boolean objectives aggregated by means of a reward function such as $\max$, cf. [Kupferman et al., 2016].
Monadic Second-Order Logic. Below we summarise MSO, which extends first-order logic with variables for sets, and recall the fundamental theorem, i.e., that MSO is decidable on regular-trees [Rabin, 1969]. The syntax of MSO includes Boolean operators ¬ and ∧; individual variables u, v, w, . . . ; set variables U, V, W, . . . ; quantifiers over these variables ∃u, ∃U; . . . ; binary relation symbols ∈, =, ≤; and unary function symbols suc_y for every d in a finite set Δ of directions. We denote formulas of MSO by Φ, Ψ, . . .

The semantics of MSO is defined over the structure TΔ = (Δ*, {suc_d}_{d∈Δ}), called the unlabelled Δ-ary tree. The interpretation of individual variables is elements in Δ*, of set variables are subsets of Δ*; Boolean operators and quantifiers are interpreted as usual; and a regular formula U is accepted by a finite automaton.

We use standard shorthands, e.g., ϵ for the root; X = Y for ∀v(v ∈ X ⇔ v ∈ Y), etc. Say that A is reachable from if for each i there is an MSO-formula φ_i(x) such that A = ϵ(A) for all MSO-formula Φ(U), we define Φ(A′) for the MSO-formula formed from Φ in which every variable U_i is replaced by the definition φ_i of A′.

Then A′ = Φ if A ≡ Φ(A′). We now introduce some abbreviations, i.e., variables for functions with finite ranges:

Definition 8 (Unary Function Variables). Let Θ be a finite set of sorts. Associate with each type θ ∈ Θ a finite set of labels L_θ. For every sort θ, we introduce unary function variables α, β, . . . of that sort, and quantification, i.e., 2α, ∃β, . . .

Define the interpretation of variable α of sort θ by a function of the form α : Δ* → L_θ. We write π to denote (TΔ, π) = Φ. If α′ is definable from α, we write Φ[α′ ↦ α] for the substitution as above.

We remark that this extension does not add expressive power. Indeed, we can replace the function variable α of sort θ by a [L_θ]-tuple of set variables X, and replace every term ω(d) by the expression v ∈ X_d.

Directions and sorts. Fix a BA-iCGS S, the direction set Δ = ACT, and the set Θ consists of four sorts:

- D with labels L_D = S_0 → S (for representing iCGS);
- H with L_H = S_0 ∪ {⊥} (for histories);
- R with L_R = S_0 → Act (for strategies);
- K with L_K = S_0 → ((Var ∪ Ag) → Act) (for assignments).

Encoding Φ(s, h, χ) by functions. Recall the bijection μ : hist(S) → S_0 × ACT* in Def. 3 and that we write h =
μ(s_h, u_h) (Section 2.2). The structure S is encoded by a function S of sort D; a history h by a function h of sort H; an assignment χ by a function χ of sort K; and a strategy σ by a function σ of sort R, as follows:

- \( \tilde{S}(v)(t) = tr(t, v) \) for all \( v \in ACT^*, t \in S_0 \).
- \( \tilde{h}(u_h) = s_h \) and \( h(v) = \bot \) for all \( v \in ACT^* \) with \( v \neq u_h \).
- \( \tilde{σ}(v)(t) = σ(μ(t, v)) \) for all \( v \in ACT^*, t \in S_0 \).
- \( \tilde{χ}(v)(t)(x) = χ(μ(t, v)) \) for all \( v \in ACT^*, t \in S_0, x \in Var \cup Ag \).

Expressing ESL in MSO. We now show how to express in MSO that a function variable α of a given sort is a valid encoding. First, we can express that a function variable α of sort H is of the form \( \tilde{h} \) for some history h, i.e., \( \exists x(α(x) ∈ S_0 ∧ ∀y(y ≠ x → (α(y) = ⊥))) \). Second, we can express that a function variable α of sort D is of the form \( \tilde{S} \), i.e., \( \bigwedge_{s\in S} (α(e(t)) = t ∧ α(sucv(e(t))) = tr(α(v)(t), d)) \). Third, for every ESL formula \( φ \), we can express that a function variable of sort K is of the form \( \tilde{χ} \) for some \( φ \)-compatible assignment \( χ \). To do this, it is sufficient to express, for \( α ∈ Ag \), that a function variable α of sort R is of the form \( \tilde{σ} \) for some strategy σ that is coherent and uniform for agent a. Coherence is easy: \( C_α(α) := ∀v \bigwedge_{s,v∈S} α(v)(s) \in Act_a \). For uniformity, we use the characterisation in Proposition 1: \( U_α(α) := \bigwedge_{s,v∈S} ∀v(E^v_a(v) → (α(v)(s) = α(v)(s'))) \) where \( E^v_a(v) \) is \( ∀w(w ≤ v → (S(w)(s) ∼_a S(w)(s'))) \).

The remainder of the proof is by structural induction.

Inductive hypothesis. For every ESL-sentence \( ϕ \) and BA-iCGS S one can construct an MSO-formula \( Φ \) such that \( (S, h, χ) \models ϕ \) if and only if \( (S, h, \tilde{χ}) \models Φ \) (for all h, χ).

Atomic predicate \( ϕ = p \). Define \( Φ \) by \( \bigvee_{s_0∈S_0, s_i∈λ-(p)} \exists v(\tilde{h}(v) = s_0 ∧ \tilde{S}(v)(s_0) = s) \).

Boolean operators. For \( ϕ = \negφ_1 \) define \( Φ = \negΦ_1 \); and for \( ϕ = φ_1 ∧ φ_2 \) define \( Φ = Φ_1 ∧ Φ_2 \).

Strategic operator \( φ = \langle x \rangle φ_1 \). Define \( Φ \) by \( \exists x \bigwedge_{s,h,x} C_α(α) ∧ U_α(α) ∧ Φ^2_1 \), where \( Φ^2_1 \) is \( Φ_1 \) in which \( \tilde{χ}(v)(s)(x) \) is replaced by \( α(v)(s) \).

Binding operator \( ϕ = (x, a)φ_1 \). Define \( Φ \) by \( Φ_1[\tilde{χ} ← χ] \) where, writing \( x' \) for \( χ_a^x(x) \), the encoding \( χ \) is definable from \( \tilde{χ} \) as follows: \( \tilde{χ}(v)(t)(y) = χ(v)(t)(y) \) if \( y ≠ a \), and equals \( χ(v)(t)(x) \) if \( y = a \).

Epistemic operator \( ϕ = K_αφ_1 \). The formula \( Φ \) is \( \bigwedge_{s,t∈S} ∀u(\tilde{h}(u) = s ∧ E^u_a(u) → Φ_1[\tilde{h}' ← h]) \), where \( h' = μ(t, u) \) (note that \( \tilde{h}' \) is definable from u and t). The other epistemic operators are treated similarly.

Next operator \( ϕ = Xφ_1 \). It is sufficient to note that, writing \( h' = π(h, |Ag|≤|h|+1) \), the encoding \( h' \) is definable from \( \tilde{χ} \) and h as follows: \( h'(v) = t \) if \( \tilde{h}(u) = t \) and \( v = suc_j(u) \) and \( J(a) = \tilde{χ}(u)(t)(a) \), else \( \tilde{χ}(v)(t)(a) \).

Until operator \( ϕ = φ_1 U φ_2 \). Note that for a) every \( s ∈ S_0 \) there is an MSO-formula \( P_s(U, u) \) that says that \( U \) is an infinite branch, \( u ∈ U \), and that after u the branch \( U \) continues

Note that for a) every \( s ∈ S_0 \) there is an MSO-formula \( P_s(U, u) \) that says that \( U \) is an infinite branch, \( u ∈ U \), and that after u the branch \( U \) continues

5 Conclusions

One of the key problems in reasoning about strategic abilities in MAS under incomplete information and perfect recall is that the model checking and synthesis problems are undecidable even for relatively weak logics such as ATL. Yet, MAS applications require specifications that are more expressive than ATL, e.g., capable of expressing solution concepts such as Nash equilibria. Identifying classes of systems for which these two desiderata can be combined remains a challenge. In this paper we have made a contribution towards this aim.

Specifically, we defined ESL, a combination of Strategy Logic and Epistemic Logic. We observed that model checking and synthesis are undecidable under synchronous perfect-recall semantics. However, we showed that a noteworthy subclass of systems, those that admit only broadcast actions, admit decidable model checking and synthesis, and identified tight bounds for the model-checking problem.

We have illustrated the expressivity of the formalism by phrasing rational synthesis under incomplete information, a previously unexplored set-up, as an instance of model checking for ESL. This has the noteworthy consequence that rational synthesis is decidable in the framework. It follows that we can decide expressive strategic properties of rational secret-sharing scenarios like the one presented in Section 3.2 under the assumption of non-randomised strategies. We leave the exploration of other scenarios for future work.

Acknowledgements

This research was partly supported by EPSRC (grant EP/00529X), INdAM (grant "Logica e Automi per il Model Checking"), the ANR JCJC (project SVeDaS) and an INdAM Marie Curie fellowship to S. Rubin. The authors thank Benjamin Aminof for fruitful discussions.
References


