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 van der 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 [van der 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 [van der Hoek and Wooldridge, 2003; Belardinelli, 2014; Huang and van der Meyden, 2014; Čermák *et al.*, 2014; **?**].

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 Δ_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, [**?**] 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; **?**]. A further important result in this area is that for broadcast systems the synthesis problem of specifications in LTL and knowledge is decidable [van der 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 ϵ. 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 $F \in \prod_i (X_i)^Y$ we may write $F(i) \in X_i^Y$ 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* $\mathsf{S} = \langle Ag, AP, \{Act_a\}_{a \in Ag}, S, S_0, \mathsf{tr}, \{\sim_a\}_{a \in Ag}, \lambda \rangle$*, 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. *$\mathsf{tr} : S \times \mathsf{ACT} \to S$ is the* transition function*, where $\mathsf{ACT} = \prod_{a \in Ag} Act_a$ 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. *$\lambda : AP \to 2^S$ 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 $a \in \mathrm{Ag}$. 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 $J_i \in \mathsf{ACT}$ such that $h_{i+1} \in \mathsf{tr}(h_i, J_i)$. The set of all histories in S is denoted as $\mathsf{hist}(\mathsf{S})$, and the set of histories h' that extend history h is denoted as $\mathsf{hist}(\mathsf{S}, h)$, that is, $h'_{\leq |h|} = h$.

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 $\mathsf{hist}(\mathsf{S})$ such that $h \equiv h'$ iff $|h| = |h'|$ (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 $h \equiv_a h'$. For $A \subseteq \mathrm{Ag}$, let $\sim_A^{\mathbb{C}} = (\cup_{a \in A} \sim_a)^*$, where $*$ denotes the reflexive and transitive closure (w.r.t. relation composition), and its extension to histories is denoted $\equiv_A^{\mathbb{C}}$. For $A \subseteq \mathrm{Ag}$, let $\sim_A^{\mathbb{D}} = \cap_{a \in A} \sim_a$, and its extension to histories is denoted $\equiv_A^{\mathbb{D}}$.

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

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 $J, J' \in \mathsf{ACT}$, if $J \neq J'$ and $s \sim_a s'$ then $\mathsf{tr}(s, J) \not\sim_a \mathsf{tr}(s', J')$. 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 $\mu : S_0 \times \mathsf{ACT}^* \to \mathsf{hist}(\mathsf{S})$ *that maps* (s_0, u) *to the history* h *of length* $|u|+1$ *and such that* $h_0 = s_0$ *and* $h_j = \mathsf{tr}(h_{j-1}, u_{j-1})$ *for* $1 \leq j \leq |u|$.

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

Lemma 1. *Let* S *be a BA-iCGS. For all* $a \in Ag$, $u, u' \in \mathsf{ACT}^*$ *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 $j < |u|$. □

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* σ *be a coherent strategy for agent* a. *Then* σ *is uniform for agent* a *if and only if for all* $v \in \mathsf{ACT}^*, s, s' \in S_0$ *we have that* $\mu(s, v) \equiv_a \mu(s', v)$ *implies* $\sigma(\mu(s, v)) = \sigma(\mu(s', v))$.

2.3 The Logic ESL

We now introduce ESL, an epistemic extension of SL. We interpret it on iCGS with history-based semantics.

Syntax. Fix a finite set of *atomic propositions (atoms)* AP, a finite set of *agents* Ag, and an infinite set Var of strategy variables $x_0, x_1, \ldots$. The *formulas over* AP, Ag, *and Var* are built according to the following grammar: $\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi \mid \langle\langle x \rangle\rangle \varphi \mid (x, a)\varphi \mid \mathbb{K}_a\varphi \mid \mathbb{C}_A\varphi \mid \mathbb{D}_A\varphi$, where $p \in AP$, $x \in$ Var, $a \in$ Ag, and $A \subseteq$ Ag. The set of ESL *formulas* is the one generated by the grammar. The *temporal operators* are X (read "next") and U (read "until"). The *strategy quantifier* is $\langle\langle x \rangle\rangle$ ("for some strategy x, ..."") and the binding operator (x, a) ("by using strategy x, agent a can enforce ..."); whilst the *epistemic operators* are $\mathbb{K}_a$ ("agent a knows that"), $\mathbb{C}_A$ ("it is common-knowledge amongst A that"), and $\mathbb{D}_A$ ("the agents in A distributively know that"). We use the usual shorthands, e.g., true for $p \vee \neg p$, $[[x]]\varphi$ for $\neg\langle\langle x \rangle\rangle\neg\varphi$, and $\mathbb{E}_A\varphi$ for $\wedge_{a \in A}\mathbb{K}_a\varphi$. The sets *free*$(\varphi)$ and *bnd*(φ) of free and bound variables appearing in a formula φ are defined as standard [Mogavero *et al.*, 2014]. Intuitively, $x \in$ *free*(φ) if x does not occur in φ within the scope of any strategy quantifier of the form $\langle\langle x \rangle\rangle$; while $a \in$ *free*(φ) if some temporal operator is not in the scope of a binding (x, a) for agent a. Hereafter we assume that sets *free*(φ) and *bnd*(φ) are disjoint. Moreover, as in [Cermák, 2014] we assume that every variable is quantified at most once in a given formula. All these properties can be ensured w.l.o.g. by renaming bound variables. A *sentence* is a formula φ with *free*$(\varphi) = \emptyset$. Finally, we define $shr(x, \varphi) = \{a \in Ag \mid (x, a)\psi$ is a subformula of $\varphi\}$ as the set of agents using strategy x in evaluating φ.

Semantics. Fix an iCGS S. An *assignment* is a function $\chi :$ Var $\cup$ Ag $\to \Sigma(\mathsf{S})$ such that for every agent $a \in$ Ag, the strategy $\chi(a)$ is coherent and uniform for a. For $x \in$ Var$\cup$Ag and $\sigma \in \Sigma(\mathsf{S})$, the *variant* χ^x_σ is the assignment that maps x to σ and coincides with χ on all other variables and agents. Moreover, if $x = a \in$ Ag then we require σ to be coherent and uniform for a. An assignment χ is φ*-compatible* if, for every $x \in$ Var, the strategy $\chi(x)$ is coherent and uniform for every agent in $shr(x, \varphi)$.

We define $(\mathsf{S}, h, \chi) \models \varphi$ where $h \in \mathsf{hist}(\mathsf{S})$, φ is a formula, χ is a φ-compatible assignment, and $\pi := \pi(h, \chi|_{\mathrm{Ag}})$ is the unique infinite sequence that extends h by following the restriction of χ to Ag:

$(\mathsf{S}, h, \chi) \models p$ iff $last(h) \in \lambda(p)$, for $p \in AP$
$(\mathsf{S}, h, \chi) \models \neg\varphi_1$ iff it is not the case that $(\mathsf{S}, h, \chi) \models \varphi_1$
$(\mathsf{S}, h, \chi) \models \varphi_1 \wedge \varphi_2$ iff $(\mathsf{S}, h, \chi) \models \varphi_i$ for $i \in \{1, 2\}$
$(\mathsf{S}, h, \chi) \models \langle\langle x \rangle\rangle\varphi_1$ iff there exists a strategy σ that is uniform and coherent for every agent in $shr(x, \varphi_1)$ such that $(\mathsf{S}, h, \chi^x_\sigma) \models \varphi_1$
$(\mathsf{S}, h, \chi) \models (x, a)\varphi_1$ iff $(\mathsf{S}, h, \chi^a_{\chi(x)}) \models \varphi_1$
$(\mathsf{S}, h, \chi) \models \mathbb{K}_a\varphi_1$ iff for every history $h' \in \mathsf{hist}(\mathsf{S})$, $h' \equiv_a h$ implies $(\mathsf{S}, h', \chi) \models \varphi_1$
$(\mathsf{S}, h, \chi) \models \mathbb{C}_A\varphi_1$ iff for every history $h' \in \mathsf{hist}(\mathsf{S})$, $h' \equiv^{\mathbb{C}}_A h$ implies $(\mathsf{S}, h', \chi) \models \varphi_1$
$(\mathsf{S}, h, \chi) \models \mathbb{D}_A\varphi_1$ iff for every history $h' \in \mathsf{hist}(\mathsf{S})$, $h' \equiv^{\mathbb{D}}_A h$ implies $(\mathsf{S}, h', \chi) \models \varphi_1$
$(\mathsf{S}, h, \chi) \models \mathsf{X}\varphi_1$ iff $(\mathsf{S}, \pi_{\leq |h|+1}, \chi) \models \varphi_1$
$(\mathsf{S}, h, \chi) \models \varphi_1 \mathsf{U} \varphi_2$ iff there exists $i \geq |h|$ s.t. $(\mathsf{S}, \pi_{\leq i}, \chi) \models \varphi_2$, for all j with $|h| \leq j < i$, $(\mathsf{S}, \pi_{\leq j}, \chi) \models \varphi_1$,

The following says that the satisfaction is well-defined:

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

For a history formula φ, we write $\mathsf{S} \models \varphi$ to mean that $(\mathsf{S}, s, \chi) \models \varphi$ for every $s \in S_0$ and assignment χ (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 χ and χ' coincide on *free*(φ), then $(\mathsf{S}, h, \chi) \models \varphi$ iff $(\mathsf{S}, h, \chi') \models \varphi$. Thus, e.g., if φ is a sentence, then $\mathsf{S}, s \models \varphi$ iff $(\mathsf{S}, s, \chi) \models \varphi$ for some assignment χ.

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

Proposition 2. *For every* SL *sentence* φ *there is an* ESL *sentence* $\hat{\varphi}$ *s.t. for all CGS* S, *we have that* $\mathsf{S} \models \varphi$ *iff* $\mathsf{S} \models \hat{\varphi}$.

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* φ *there is an* ESL *sentence* $\hat{\varphi}$ *s.t. for all iCGS* S, *we have that* $\mathsf{S} \models \varphi$ *iff* $\mathsf{S} \models \hat{\varphi}$.

Proof. The main difficulty is to translate the ATL* operator $\langle\langle A \rangle\rangle$ in ESL, which we illustrate. Consider Ag $= \{a_1, a_2, a_3, a_4\}$, $A = \{1, 2\}$, and the ATL*K formula $\varphi = \langle\langle A \rangle\rangle\psi$. The formula says that for every set of uniform strategies, one for each agent in A, every path consistent with these strategies satisfies ψ. Consider the ESL formula $\hat{\varphi} = \langle\langle x_1 \rangle\rangle\langle\langle x_2 \rangle\rangle [[x_3]][[x_4]](x_1, a_1)(x_2, a_2)(x_3, a_3)(x_4, a_4)\hat{\psi}$. Clearly, φ logically implies $\hat{\varphi}$ 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 π be any path consistent with strategies σ_1, σ_2 (for agents a_1, a_2). It is sufficient to show that there exist uniform strategies, σ_3 for agent a_3 and σ_4 for agent a_4, such that $\pi(s, \sigma_{\mathrm{Ag}}) = \pi$. Indeed, for every $n \geq 0$, let $J \in \mathsf{ACT}$ be a joint action such that i) $\sigma_i(\pi_{\leq n}) = J(i)$ for $i = 1, 2$, and ii) $\mathsf{tr}(\pi_n, J) = \pi_{n+1}$. Define $\sigma_i(\pi_{\leq n}) = J(i)$ for $i = 3, 4$. Since the uniformity condition only restricts pairs of histories of the same length, we can extend σ_3 and σ_4 to uniform strategies. Note that $\pi(s, \sigma_{\mathrm{Ag}}) = \pi$, as required. □

We now introduce the main decision problem of this work.

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

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 $\mathcal{C}$ consists of all iCGS with $|\mathrm{Ag}| = 3$ and $\mathcal{F}$ contains just the ATL formula $\langle\langle\{1, 2\}\rangle\rangle \mathsf{G}\, p$, 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 amongst 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 [**?**]. Technically, suppose we are given an iCGS S representing the multi-agent system, and LTLK-formulas γ_a representing the objective of agent $a \in \mathrm{Ag}$. Here, LTLK is the logic consisting of the set of path-formulas of $\mathsf{ATL}^*\mathsf{K}$. We can then talk about Nash equilibria $\overline{\sigma}$ in games of the form $G = \langle \mathsf{S}, \{\gamma_a\}_{a \in \mathrm{Ag}} \rangle$[1]. Rational synthesis considers the following decision problem (sometimes called *E-NASH*) :

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

Intuitively, φ represents some global property that the designer wants to ensure given that agents are self-interested. In case $\varphi = \mathsf{true}$, 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 $\overline{\sigma}$. The dual problem, called *Strong Rational Synthesis* (sometimes called *A-NASH*), concerns deciding whether all Nash equilibria induce a path that satisfies φ [Kupferman *et al.*, 2016].

We now show that rational synthesis for LTLK objectives reduces to model checking against ESL. Suppose $\mathrm{Ag} = \{a_1, a_2, \ldots, a_n\}$, and let $\overline{x}$ be an n-tuple of variables. Let β 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 $\mathsf{RatSyn}_{\varphi}(\overline{x})$ in ESL expresses that $\overline{x}$ is a Nash equilibrium whose induced execution satisfies φ: $(x_1, a_1) \ldots (x_n, a_n) \left[\varphi \wedge \bigwedge_{a \in \mathrm{Ag}} (\langle\langle y \rangle\rangle (y, a) \gamma_a \rightarrow \gamma_a) \right]$.

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

Lemma 3. *Rational synthesis for* LTLK *objectives is reducible to model checking against the* ESL*-formula $\langle\langle x_1 \rangle\rangle \ldots \langle\langle x_n \rangle\rangle \mathsf{RatSyn}_{\varphi}(\overline{x})$.*

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 $\mathrm{Ag} = \{1, 2, \ldots, n\}$, initially each agent $i \in \mathrm{Ag}$ privately holds a "share" f_i of a secret f_0, and any m "good" agents can collaborate to learn the secret

[1]The 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].

in spite of the remaining $n-m$ "bad" agents[2]. In the rational version of this scenario, the objective of each agent is to learn the secret, i.e., she prefers to learn the secret rather than not to learn it. Richer, non-binary, preferences can also be handled, including the fact that an agent may prefer that the least number of other agents learn the secret. For simplicity we do not consider such extensions here.

We can model this scenario as an iCGS as follows. The secret is the value of a variable s initially hidden from all agents (formally, a variable v with finite domain D is modelled as $|D|$-many atomic propositions); agent i's share is modelled as a private variable f_i; each agent has a private variable s_i that represents what she thinks the secret is; at every step, every agent broadcasts a message (from some fixed finite set of M messages). Finally, the objective γ_i of each agent i can be formalised as the LTLK-formula $\mathsf{F}\,\mathsf{G}\,\mathbb{K}_i(s_i = s)$: *from some point on, agent i knows the secret*. Thus, the ESL-formula $\langle\!\langle x_1\rangle\!\rangle \dots \langle\!\langle x_n\rangle\!\rangle \mathsf{RatSyn}_{\varphi}(\overline{x})$ expresses that there is a Nash equilibrium satisfying φ in the rational secret-sharing scenario. For instance, one can use φ to express that agents make "true" statements, e.g., that if agent i broadcasts "my share is x", then indeed $f_i = x$. Observe that by using ESL specifications we can naturally express secrecy and strategic concepts.

4 Model Checking BA-iCGS against ESL

In this section we prove the main technical result of this paper.

Theorem 6. *Model checking BA-iCGS against* ESL *specifications is decidable and non-elementary complete.*

For the non-elementary lower-bound we use the observation that model-checking SL on CGS (i.e., with perfect-information) is non-elementary [Mogavero *et al.*, 2014], together with the fact that by encoding the last joint action into the states, one can translate a CGS S into a BA-iCGS S' such that for all sentences φ in ESL, we have that $\mathsf{S} \models \varphi$ iff $\mathsf{S}' \models \varphi$ (the same procedure is used in [Belardinelli *et al.*, 2017]).

For the non-elementary upper-bound, we reduce the model-checking problem of BA-iCGS against ESL specifications to model checking regular-trees against Monadic Second-Order Logic (MSO). The naïve approach is to code every tuple (S, h, χ) by a tuple of functions $(\widehat{\mathsf{S}}, \widehat{h}, \widehat{\chi})$ each of whose domain is the set ACT^* of finite sequences of joint actions, and whose ranges are finite (to be specified later). This encoding allows us to build, for every ESL-sentence φ, an MSO-formula Φ, such that $(\mathsf{S}, h, \chi) \models \varphi$ iff $T \models \Phi(\widehat{\mathsf{S}}, \widehat{h}, \widehat{\chi})$, where T is the infinite ACT-ary tree generated by $\widehat{\mathsf{S}}$, $\widehat{h}$, and $\widehat{\chi}$. The latter problem is decidable if $\widehat{\mathsf{S}}$, $\widehat{h}$, and $\widehat{\chi}$ are regular functions (a function $f : D^* \to L$ is *regular* if, for each $l \in L$, the set $f^{-1}(l) \subseteq D^*$ is accepted by a finite automaton). Since φ is a sentence we can choose χ arbitrarily, in particular so that it is regular (on the other hand, both $\widehat{\mathsf{S}}$ and $\widehat{h}$ are always regular).

[2]In Shamir's scheme this is implemented by an initially unknown polynomial f of degree $m-1$ in some finite field F with $|F| > n$ and $f(0) \neq 0$; the secret is $f(0)$, each share is $f(i)$, and thus any m shares uniquely determine the secret (by interpolation).

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 $\neg$ and $\wedge$; individual variables $u, v, w, \dots$; set variables $U, V, W, \dots$; quantifiers over these variables $\exists u, \exists U, \dots$; binary relation symbols $\in$, $=$, and $\preceq$; and unary function symbols suc_d for every d in a finite set Δ of *directions*. We denote formulas of MSO by $\Phi, \Psi, \dots$. The *semantics* of MSO is defined over the structure $\mathsf{T}_\Delta = \langle \Delta^*, \{\mathsf{suc}_d\}_{d\in\Delta}\rangle$, called the *unlabelled Δ-ary tree*. The interpretation of individual variables are elements in Δ^*, of set variables are subsets of Δ^*; Boolean operators and quantifiers are interpreted as usual; atoms $u \in U$ and $u = v$ as usual; while $u \preceq v$ is the prefix relation, and $\mathsf{suc}_d(u) = ud$ for $d \in \Delta, u \in \Delta^*$. We will often think of $u \in \Delta^*$ as the singleton set $U = \{u\} \subseteq \Delta^*$. Formulas $\Phi(\overline{U})$ with free variables $\overline{U}$ are interpreted in expanded structures $(\mathsf{T}_\Delta, \overline{A})$, called *labelled Δ-ary trees*, where each $A_i \subseteq \Delta^*$. Instead of writing $(\mathsf{T}_\Delta, \overline{A}) \models \Phi(\overline{U})$, we may write $\mathsf{T}_\Delta \models \Phi(\overline{A})$, or simply $\overline{A} \models \Phi$. A labelled-tree $(\mathsf{T}_\Delta, \overline{A})$ is *regular* if each $A_i \subseteq \Delta^*$ is accepted by a finite automaton.

Theorem 7. *[Rabin, 1969] There is a non-elementary time algorithm that, given an* MSO*-formula $\Phi(\overline{U})$ and a regular labelled-tree $(\mathsf{T}_\Delta, \overline{A})$, decides whether $\mathsf{T}_\Delta \models \Phi(\overline{A})$. Also, if $\mathsf{T}_\Delta \models \exists \overline{U}\, \Phi(\overline{U})$ then there is a regular labelled-tree $(\mathsf{T}_\Delta, \overline{A})$ such that $\mathsf{T}_\Delta \models \Phi(\overline{A})$, and the finite automata for all $A_i \subseteq \Delta^*$ are computable.*

We use standard shorthands, e.g., ϵ for the root; $X = Y$ for $\forall v(v \in X \leftrightarrow v \in Y)$, etc. Say that $\overline{A'}$ *is definable from* $\overline{A}$ if for each i there is an MSO-formula $\varphi_i(x)$ such that $\overline{A} \models \forall x(\varphi_i(x) \leftrightarrow x \in A'_i)$. For an MSO-formula $\Phi(\overline{U})$, we define $\Phi[\overline{A'} \leftarrow \overline{A}]$ for the MSO-formula formed from Φ in which every variable U_i is replaced by the definition φ_i of A'_i. Then $\overline{A'} \models \Phi$ iff $\overline{A} \models \Phi[\overline{A'} \leftarrow \overline{A}]$. 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 $\theta \in \Theta$ a* finite *set of* labels *L_θ. For every sort θ, we introduce unary function variables $\alpha, \beta, \dots$ of that sort, and quantification, i.e, $\exists\alpha, \exists\beta, \dots$. Define the interpretation of variable α of sort θ by a function of the form $\alpha : \Delta^* \to L_\theta$. We write $\overline{\alpha} \models \Phi$ to denote $(\mathsf{T}_\Delta, \overline{\alpha}) \models \Phi$. If α' is definable from α, we write $\Phi[\alpha' \leftarrow \alpha]$ 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 $|L_\theta|$-tuple of set variables $\overline{X}$, and replace every term $\alpha(v) = d$ by the expression $v \in X_d$.

Directions and sorts. Fix a BA-iCGS S, the direction set $\Delta = \mathsf{ACT}$, and the set Θ to consists of four sorts:

- D with labels $L_D = S_0 \to S$ (for representing iCGS);
- H with $L_H = S_0 \cup \{\bot\}$ (for histories);
- R with $L_R = S_0 \to$ Act (for strategies);
- K with $L_K = S_0 \to ((\text{Var} \cup \text{Ag}) \to \text{Act})$ (for assignments).

Encoding (S, h, χ) by functions. Recall the bijection $\mu : \mathsf{hist}(\mathsf{S}) \to S_0 \times \mathsf{ACT}^*$ in Def. 3 and that we write $h =$

$\mu(s_h, u_h)$ (Section 2.2). The structure S is encoded by a function $\widehat{\mathsf{S}}$ of sort D; a history h by a function $\widehat{h}$ of sort H; an assignment χ by a function $\widehat{\chi}$ of sort K; and a strategy σ by a function $\widehat{\sigma}$ of sort R, as follows[3]:

- $\widehat{\mathsf{S}}(v)(t) = \mathsf{tr}(t, v)$ for all $v \in \mathsf{ACT}^*, t \in S_0$.
- $\widehat{h}(u_h) = s_h$, and $\widehat{h}(v) = \bot$ for all $v \in \mathsf{ACT}^*$ with $v \neq u_h$.
- $\widehat{\sigma}(v)(t) = \sigma(\mu(t, v))$ for all $v \in \mathsf{ACT}^*, t \in S_0$.
- $\widehat{\chi}(v)(t)(x) = \chi(x)(\mu(t, v))$ for all $v \in \mathsf{ACT}^*$, $t \in S_0$, $x \in \mathrm{Var} \cup \mathrm{Ag}$.

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 $\widehat{h}$ for some history h, i.e., $\exists x(\alpha(x) \in S_0 \wedge \forall y(y \neq x \rightarrow (\alpha(y) = \bot)))$. Second, we can express that a function variable α of sort D is of the form $\widehat{\mathsf{S}}$, i.e., $\bigwedge_{t \in S}(\alpha(\epsilon)(t) = t \wedge \alpha(\mathsf{suc}_d(v))(t) = \mathsf{tr}(\alpha(v)(t), d))$. Third, for every ESL formula φ, we can express that a function variable of sort K is of the form $\widehat{\chi}$ for some φ-compatible assignment χ. To do this, it is sufficient to express, for $a \in Ag$, that a function variable α of sort R is of the form $\widehat{\sigma}$ for some strategy σ that is coherent and uniform for agent a. Coherency is easy: $C_a(\alpha) := \forall v \bigwedge_{s \in S_0} \alpha(v)(s) \in \mathrm{Act}_a$. For uniformity, we use the characterisation in Proposition 1: $U_a(\alpha) := \bigwedge_{s,s' \in S_0} \forall v \left(E^a_{s,s'}(v) \rightarrow (\alpha(v)(s) = \alpha(v)(s'))\right)$ where $E^a_{s,s'}(v)$ is $\forall w(w \preceq v \rightarrow (\widehat{\mathsf{S}}(w)(s) \sim_a \widehat{\mathsf{S}}(w)(s')))$.

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 $(\mathsf{S}, h, \chi) \models \varphi$ if and only if $(\widehat{\mathsf{S}}, \widehat{h}, \widehat{\chi}) \models \Phi$ (for all h, χ).

Atomic predicate $\varphi = p$. Define Φ by $\bigvee_{s_0 \in S_0, s \in \lambda^{-1}(p)} \exists v(\widehat{h}(v) = s_0 \wedge \widehat{S}(v)(s_0) = s)$.

Boolean operators. For $\varphi = \neg\varphi_1$ define $\Phi = \neg\Phi_1$; and for $\varphi = \varphi_1 \wedge \varphi_2$ define $\Phi = \Phi_1 \wedge \Phi_2$.

Strategic operator $\varphi = \langle\langle x \rangle\rangle \varphi_1$. Define Φ by $\exists \alpha \bigwedge_{a \in shr(x,\varphi)}(C_a(\alpha) \wedge U_a(\alpha) \wedge \Phi'_1)$, where Φ'_1 is Φ_1 in which $\widehat{\chi}(v)(s)(x)$ is replaced by $\alpha(v)(s)$.

Binding operator $\varphi = (x, a)\varphi_1$. Define Φ by $\Phi_1[\widehat{\chi'} \leftarrow \widehat{\chi}]$ where, writing χ' for $\chi^a_{\chi(x)}$, the encoding $\widehat{\chi'}$ is definable from $\widehat{\chi}$ as follows: $\widehat{\chi'}(v)(t)(y)$ equals $\widehat{\chi}(v)(t)(y)$ if $y \neq a$, and equals $\widehat{\chi}(v)(t)(x)$ if $y = a$.

Epistemic operator $\varphi = \mathbb{K}_a\varphi_1$. The formula Φ is $\bigwedge_{s,t \in S_0} \forall u \left((\widehat{h}(u) = s \wedge E^a_{s,t}(u)) \rightarrow \Phi_1[\widehat{h'} \leftarrow \widehat{h}]\right)$, where $h' = \mu(t, u)$ (note that $\widehat{h'}$ is definable from u and t). The other epistemic operators are treated similarly.

Next operator $\varphi = \mathsf{X}\varphi_1$. It is sufficient to note that, writing $h' = \pi(h, \chi|_{\mathrm{Ag}})_{\leq |h|+1}$, the encoding $\widehat{h'}$ is definable from $\widehat{\chi}$ and $\widehat{h}$ as follows: $\widehat{h'}(v) = t$ if $\widehat{h}(u) = t$ and $v = \mathsf{suc}_J(u)$ and $J(a) = \widehat{\chi}(u)(t)(a)$, else $\widehat{h}(v) = \bot$.

Until operator $\varphi = \varphi_1 \mathsf{U} \varphi_2$. Note that a) for every $s \in S_0$ there is an MSO-formula $P_s(U, u)$ that says that U is an infinite branch, $u \in U$, and that after u the branch U continues by following the joint full strategy induced by $\widehat{\chi}$ from initial state s; and b) $\mu(s, u) \in \mathsf{out}(\mathsf{S}, h)$ can be expressed in terms of $\widehat{h}$ by $\exists x(u \preceq x \wedge \widehat{h}(x) = s)$. Thus, the translation of $\mathsf{F}\varphi_2 \equiv \mathsf{true}\,\mathsf{U}\,\varphi_2$ (the full operator U is similar) is $\bigvee_{s \in S_0} \exists U \exists u(P_s(U, u) \wedge \widehat{h}(u) = s \wedge \exists \widehat{h'} \exists v \in U(u \preceq v \wedge \widehat{h'}(v) = s \wedge \Phi_2[\widehat{h'} \leftarrow \widehat{h}]))$. This completes the induction.

The size of Φ is polynomial in the size of the input (i.e., φ, S). Applying Theorem 7 we get the stated non-elementary upper-bound. This completes the proof of Theorem 6.

Application to Rational Synthesis. By the discussion in Section 3.1, we immediately get the first part of the following:

Corollary 1. *Rational synthesis for* LTLK *objectives on BA-iCGS is decidable. Moreover, if a given instance returns "yes", then a finite-state Nash equilibrium can be computed.*

For the second part, apply the translation presented above to the ESL-formula $\mathsf{RatSyn}_{\mathsf{true}}(\overline{x})$ and a BA-iCGS S (say with $|Ag| = n$) to get an MSO-formula $\Phi(\overline{U})$ such that for all strategy profiles $\overline{\sigma}$, we have that $\mathsf{S} \models \mathsf{RatSyn}_{\mathsf{true}}(\overline{x})$ iff $(\widehat{\sigma_1}, \ldots, \widehat{\sigma_n}) \models \Phi$. Now, by Theorem 7 applied to Φ, one can compute regular languages $A_i \subseteq \mathsf{ACT}^*$ such that $\overline{A} \models \Phi$. Since these languages code strategies, we have computed finite-state strategies σ_i such that $\mathsf{S} \models \mathsf{RatSyn}_{\mathsf{true}}(\overline{\sigma})$.

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/I00529X), 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.

[3]Hereafter $\mathsf{tr} : S_0 \times \mathsf{ACT}^* \rightarrow S$ is defined by $\mathsf{tr}(s, \epsilon) = s$ and $\mathsf{tr}(s, vx) = \mathsf{tr}(\mathsf{tr}(s, v), x)$ for $v \in \mathsf{ACT}^*, x \in \mathsf{ACT}$.

References

[Abraham *et al.*, 2006] I. Abraham, D. Dolev, R. Gonen, and J. Y. Halpern. Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In *PODC'06*, pages 53–62, 2006.

[Abraham *et al.*, 2011] I. Abraham, L. Alvisi, and J.Y. Halpern. Distributed computing meets game theory: combining insights from two fields. *SIGACT News*, 42(2):69–76, 2011.

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

[Belardinelli and Lomuscio, 2009] F. Belardinelli and A. Lomuscio. Quantified Epistemic Logics for Reasoning About Knowledge in Multi-Agent Systems. *Artificial Intelligence*, 173(9-10):982–1013, 2009.

[Belardinelli *et al.*, 2017] F. Belardinelli, A. Lomuscio, A. Murano, and S. Rubin. Verification of multi-agent systems with imperfect information and public actions. In *AAMAS'17*, 2017.

[Belardinelli, 2014] F. Belardinelli. Reasoning about knowledge and strategies: Epistemic strategy logic. In *SR'14*, EPTCS 146, pages 27–33, 2014.

[Bulling and Jamroga, 2014] N. Bulling and W. Jamroga. Comparing variants of strategic ability: how uncertainty and memory influence general properties of games. *JAAMAS*, 28(3):474–518, 2014.

[Čermák *et al.*, 2014] P. Čermák, A. Lomuscio, F. Mogavero, and A. Murano. MCMAS-SLK: A Model Checker for the Verification of Strategy Logic Specifications. In *CAV'14*, LNCS 8559, pages 524–531. Springer, 2014.

[Cermák, 2014] P. Cermák. A model checker for strategy logic. Master's thesis, Dep. of Computing, Imperial, 2014.

[Clarke *et al.*, 2002] E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking.* MIT Press, 2002.

[Dima and Tiplea, 2011] C. Dima and F.L. Tiplea. Model-checking ATL under imperfect information and perfect recall semantics is undecidable. *CoRR*, abs/1102.4225, 2011.

[Fagin *et al.*, 1995] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning about Knowledge.* MIT, 1995.

[Fisman *et al.*, 2010] D. Fisman, O. Kupferman, and Y. Lustig. Rational Synthesis. In *TACAS'10*, LNCS 6015, pages 190–204. Springer, 2010.

[Gutierrez *et al.*, 2016] J. Gutierrez, G. Perelli, and M. Wooldridge. Imperfect information in reactive modules games. In *Principles of Knowledge Representation and Reasoning: Proceedings of the 15th International Conference*, pages 390–400. AAAI Press, 2016.

[Gutierrez *et al.*, 2017] J. Gutierrez, P. Harrenstein, and M. Wooldridge. Reasoning about equilibria in game-like concurrent systems. *Annals of Pure and Applied Logic*, 168(2):373–403, 2017.

[Halpern and Vardi, 1989] J. Halpern and M. Vardi. The complexity of reasoning about knowledge and time. I. Lower bounds. *JCSS*, 38(1):195–237, 1989.

[Halpern *et al.*, 2003] J. Halpern, R. van der Meyden, and M. Y. Vardi. Complete axiomatisations for reasoning about knowledge and time. *SIAM J. Comp.*, 33(3):674–703, 2003.

[Huang and van der Meyden, 2014] X. Huang and R. van der Meyden. A temporal logic of strategic knowledge. In *KR'14*. AAAI Press, 2014.

[Jamroga and Dix, 2006] W. Jamroga and J. Dix. Model checking abilities under incomplete information is indeed Δ_p^2-complete. In *EUMAS'06*, pages 14–15, 2006.

[Jamroga and van der Hoek, 2004] W. Jamroga and W. van der Hoek. Agents that know how to play. *Fundamenta Informaticae*, 62:1–35, 2004.

[Kupferman *et al.*, 2016] O. Kupferman, G. Perelli, and M.Y. Vardi. Synthesis with rational environments. *Ann. Math. Artif. Intell.*, 78(1):3–20, 2016.

[Meyer and van der Hoek, 1995] J.-J. Ch. Meyer and W. van der Hoek. *Epistemic Logic for AI and Computer Science*. Cambridge University Press, 1995.

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

[Pnueli and Rosner, 1989] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proc. of POPL 1989*, pages 179–190, 1989.

[Rabin, 1969] M.O. Rabin. Decidability of Second-Order Theories and Automata on Infinite Trees. *TAMS*, 141:1–35, 1969.

[Sandholm, 1999] T.W. Sandholm. Distributed rational decision making. *Multiagent systems: a modern approach to distributed artificial intelligence*, pages 201–258, 1999.

[van der Hoek and Wooldridge, 2003] W. van der Hoek and M. Wooldridge. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75(1):125–157, 2003.

[van der Meyden and Shilov, 1999] R. van der Meyden and H. Shilov. Model checking knowledge and time in systems with perfect recall. In *FST&TCS'99*, pages 432–445, 1999.

[van der Meyden and Wilke, 2005] R. van der Meyden and T. Wilke. Synthesis of distributed systems from knowledge-based specifications. In *CONCUR'05*, LNCS 3653, pages 562–576. Springer, 2005.

[Wooldridge *et al.*, 2016] M. Wooldridge, J. Gutierrez, P. Harrenstein, E. Marchioni, G. Perelli, and A. Toumi. Rational verification: From model checking to equilibrium checking. In *AAAI'16*, pages 4184–4191, 2016.