The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)

An Abstraction-Based Method for Verifying Strategic Properties in Multi-Agent Systems with Imperfect Information

Francesco Belardinelli
Imperial College London, UK
Laboratoire IBISC,
Université d'Evry, France

Alessio Lomuscio
Imperial College London
UK

Vadim Malvone
Laboratoire IBISC,
Université d'Evry
France

Abstract

We investigate the verification of Multi-agent Systems against strategic properties expressed in Alternating-time Temporal Logic under the assumptions of imperfect information and perfect recall. To this end, we develop a three-valued semantics for concurrent game structures upon which we define an abstraction method. We prove that concurrent game structures with imperfect information admit perfect information abstractions that preserve three-valued satisfaction. Further, we present a refinement procedure to deal with cases where the value of a specification is undefined. We illustrate the overall procedure in a variant of the Train Gate Controller scenario under imperfect information and perfect recall.

1 Introduction

Alternating-time Temporal Logic (ATL) and its extension ATL^* are well-known formalisms for reasoning about strategic bahaviours in Multi-agent Systems (Alur, Henzinger, and Kupferman 2002). An attractive feature of ATL is the computational complexity of its model checking problem, which is PTIME-complete under the assumption of perfect information. Multi-agent systems (MAS), however, typically exhibit imperfect information, and model checking MAS against ATL specifications under imperfect information and perfect recall is known to be undecidable (Dima and Tiplea 2011). Given the practical and theoretical importance of the imperfect information setting, even partial solutions to the problem can be useful. Previous approaches (see related work below) have either focused on how the information is shared amongst the agents in the system (Belardinelli et al. 2017b; 2017a), or developed notions of bounded recall (Belardinelli, Lomuscio, and Malvone 2018).

Instead, at the heart of the present contribution is the idea that, under a three-valued semantics, MAS with imperfect information can be approximated (or *abstracted*) by perfect information variants. This enables us to derive a sound, albeit incomplete, verification procedure for ATL and ATL^* under imperfect information and perfect recall. In more detail, given a concurrent game structure with imperfect information (iCGS) representing a MAS, we build a perfect information abstraction that preserves satisfaction for a three-valued variant of ATL^*. As we show, if the ATL^* specification is true (resp. false) in the (perfect information) abstraction, then it is also true (resp. false) in the original iCGS with imperfect information. On the other hand, if the specification is undefined, we can proceed to refining the abstraction in an attempt to give a defined truth value to the specification. The original problem is undecidable; so no guarantee can be given that by succesive refinements, the property's truth or falsity can ever be established. However, the procedure provides a constructive method to partially model check ATL^* under imperfect information and perfect recall.

Related work. Several approaches for the verification of specifications in ATL and ATL^* under imperfect information and perfect recall have been recently put forward. In one line, restrictions are made on how information is shared amongst the agents, so as to retain decidability (Berthon et al. 2017). In a related line, interactions amongst agents are limited to public actions only (Belardinelli et al. 2017b; 2017a). These approaches are markedly different from ours as they seek to identify classes for which verification is decidable. Instead, we consider the whole class of iCGS and define a general verification procedure. In this sense, our approach is closely related to (Belardinelli, Lomuscio, and Malvone 2018) where a bounded recall method, also incomplete, is defined. However, while in that work perfect recall is approximated, here abstraction is carried out on the levels of information.

At the heart of the method we describe is the notion of abstraction and refinement of MAS models, as well as three-valued semantics in modal languages. An abstraction-refinement framework for CTL over the 3-valued semantics was studied in (Shoham and Grumberg 2004; 2007) and the case of hierarchical systems is considered in (Aminof, Kupferman, and Murano 2012). Moreover, in (Grumberg et al. 2007) an abstraction-refinement technique for full μ-calculus is introduced. An abstraction-refinement procedure for network games with perfect information was introduced in (Avni, Guha, and Kupferman 2017) and a symbolic abstraction-refinement approach to the solution of two-player games with reachability or safety goals is shown in (de Alfaro and Roy 2010). Games with incomplete information are studied in (Dimitrova and Finkbeiner 2008) by considering only safety goals and, as we do in this paper, abstraction and refinement are used to generate from a game with imperfect information a new one with perfect information. Model checking MAS by abstraction in an epistemic context was originally investigated in (Cohen et al. 2009; Belardinelli and Lomuscio 2016). Three-valued abstractions for the verification of ATL properties have also been put forward in (Ball and Kupferman 2006; Lomuscio and Michaliszyn 2014; 2015; 2016). There are, however, considerable differences between these approaches and the one here pursued. In fact, the methods above focus on decidable settings. In (Ball and Kupferman 2006; Shoham and Grumberg 2004) ATL^* is interpreted under perfect information; while (Lomuscio and Michaliszyn 2014; 2015; 2016) considers *non-uniform* strategies (Raimondi and Lomuscio 2005). In both cases the corresponding model checking problem is decidable. Their aim, therefore, is to speed-up the verification task and not, as we do here, to provide a sound procedure for an undecidable problem. Finally, in (Jamroga, Konikowska, and Penczek 2016) is shown a multi-valued semantics for ATL^* that is a conservative extension of the classical 2-valued variant. Mainly, they consider the model checking problem for perfect information games but they also refer at the imperfect information case by giving an undecidable result in general and an exponential-time result for singleton coalitions.

2 Classic Imperfect Information

In this section we introduce a two-valued semantics for the Alternating-time Temporal Logic ATL^* under imperfect information and perfect recall. To fix the notation, we assume that $Ag = \{1, \ldots, m\}$ is the set of agents and AP the set of atomic propositions. Given a set U, $\overline{U}$ denotes its complement. We denote the length of a tuple v as $|v|$, and its i-th element either as v_i or $v.i$. Let $last(v) = v_{|v|}$ be the last element in v. For $i \leq |v|$, let $v_{\geq i}$ be the suffix $v_i, \ldots, v_{|v|}$ of v starting at v_i and $v_{\leq i}$ the prefix $v_1, \ldots, v_i$ of v.

Models for MAS. We begin by giving a formal model for Multi-agent Systems by means of concurrent game structures with imperfect information (Alur, Henzinger, and Kupferman 2002; Jamroga and van der Hoek 2004).

Definition 1 (iCGS)**.** *Given sets Ag of agents and AP of atoms, a* concurrent game structure with imperfect information (iCGS) *is a tuple $M = \langle Ag, AP, S, s_0, \{Act_i\}_{i \in Ag}, d, \delta, \{\sim_i\}_{i \in Ag}, V\rangle$ such that:*

- *$S \neq \emptyset$ is a finite set of* states*, with* initial state *$s_0 \in S$.*
- *For every $i \in Ag$, Act_i is a nonempty finite set of* actions*. Let $Act = \bigcup_{i \in Ag} Act_i$ be the set of all actions, and $ACT = \prod_{i \in Ag} Act_i$ the set of all joint actions.*
- *The* protocol function *$d : Ag \times S \to (2^{Act} \setminus \emptyset)$ defines the availability of actions so that for every $i \in Ag$, $s \in S$, (i) $d(i, s) \subseteq Act_i$ and (ii) $s \sim_i s'$ implies $d(i, s) = d(i, s')$.*
- *The* (deterministic) transition function *$\delta : S \times ACT \to S$ assigns a successor state $s' = \delta(s, \vec{a})$ to each state $s \in S$, for every joint action $\vec{a} \in ACT$ such that $a_i \in d(i, s)$ for every $i \in Ag$, that is, $\vec{a}$ is* enabled *at s.*
- *For every $i \in Ag$, $\sim_i$ is a relation of* indistinguishability *between states. That is, given states $s, s' \in S$, $s \sim_i s'$ iff s and s' are observationally indistinguishable for agent i.*
- *$V : S \times AP \to \{\mathrm{tt}, \mathrm{ff}\}$ is the* two-valued labelling function*.*

By Def. 1 an iCGS describes the interactions of a group Ag of agents, starting from the initial state $s_0 \in S$, according to the transition function δ. The latter is constrained by the availability of actions to agents, as specified by the protocol function d. Further, we assume that every agent i has imperfect information of the exact state of the system; so in any state s, i considers epistemically possible all states s' that are i-indistinguishable from s (Fagin et al. 1995). When every $\sim_i$ is the identity relation, i.e., $s \sim_i s'$ iff $s = s'$, we obtain a standard CGS with perfect information (Alur, Henzinger, and Kupferman 2002). Hereafter we consider both the class $iCGS$ of all iCGS, and its subclass CGS of all CGS with perfect information.

Given a set $\Gamma \subseteq Ag$ of agents and a joint action $\vec{a} \in ACT$, let $\vec{a}_\Gamma$ and $\vec{a}_{\overline{\Gamma}}$ be two tuples comprising only of actions for the agents in Γ, resp. $\overline{\Gamma}$. We also write $\vec{a}_i$ and $\vec{a}_{\overline{i}}$ for $\vec{a}_{\{i\}}$ and $\vec{a}_{\overline{\{i\}}}$ respectively. Finally, for $\vec{a}$ and $\vec{b}$ in ACT, $(\vec{a}_\Gamma, \vec{b}_{\overline{\Gamma}})$ denotes the joint action where the actions for the agents in Γ (resp. $\overline{\Gamma}$) are taken from $\vec{a}$ (resp. $\vec{b}$).

A history $h \in S^+$ is a finite (non-empty) sequence of states. The indistinguishability relations are extended to histories in a synchronous, pointwise way, i.e., histories $h, h' \in S^+$ are *indistinguishable* for agent $i \in Ag$, or $h \sim_i h'$, iff (i) $|h| = |h'|$ and (ii) for all $j \leq |h|$, $h_j \sim_i h'_j$.

Syntax. To reason about the strategic abilities of agents in iCGS with imperfect information, we use the Alternating-time Temporal Logic ATL^* (Alur, Henzinger, and Kupferman 2002).

Definition 2 (ATL^*)**.** *State (φ) and path (ψ) formulas in ATL^* are defined as follows, where $q \in AP$ and $\Gamma \subseteq Ag$:*

$$\varphi ::= q \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle\Gamma\rangle\rangle\psi$$
$$\psi ::= \varphi \mid \neg\psi \mid \psi \wedge \psi \mid X\psi \mid (\psi U \psi)$$

Formulas in ATL^* are all and only the state formulas.

As customary, a formula $\langle\langle\Gamma\rangle\rangle\Phi$ is read as "the agents in coalition Γ have a strategy to achieve Φ". The meaning of linear-time operators *next* X and *until* U is standard (Baier and Katoen 2008). Operators $[\![\Gamma]\!]$, *release* R, *finally* F, and *globally* G can be introduced as usual.

Formulas in the ATL fragment of ATL^* are obtained from Def. 2 by restricting path formulas ψ as follows, where φ is a state formula and R is the *release* operator:

$$\psi ::= X\varphi \mid (\varphi U \varphi) \mid (\varphi R \varphi)$$

Hereafter we also consider the fragment of Γ-formulas, i.e., formulas in which the strategic operator $\langle\langle\Gamma\rangle\rangle$ ranges only over some coalition $\Gamma \subseteq Ag$.

Semantics. When giving a semantics to ATL^* formulas we assume that agents are endowed with *uniform strategies* (Jamroga and van der Hoek 2004), i.e., they perform the same action whenever they have the same information.

Definition 3 (Uniform Strategy with Perfect Recall)**.** *A uniform strategy with perfect recall for agent $i \in Ag$ is a function $f_i : S^+ \to Act_i$ such that for all histories $h, h' \in S^+$, (i) $f_i(h) \in d(i, last(h))$; and (ii) if $h \sim_i h'$ then $f_i(h) = f_i(h')$.*

By Def. 3 any strategy for agent i has to return actions that are enabled for i. Also, whenever two histories are indistinguishable for i, then the same action is returned. Notice that, for the case of (perfect information) CGS, condition (ii) is satisfied by any strategy $f_i : S^+ \to Act_i$.

Given an iCGS M, a *path* $p \in S^\omega$ is an infinite sequence $s_1 s_2 \dots$ of states. Given a joint strategy $F_\Gamma = \{f_i \mid i \in \Gamma\}$, comprising of one strategy for each agent in coalition Γ, a path p is F_Γ-*compatible* iff for every $j \geq 1$, $p_{j+1} = \delta(p_j, \vec{a})$ for some joint action $\vec{a}$ such that for every $i \in \Gamma$, $a_i = f_i(p_{\leq j})$, and for every $i \in \overline{\Gamma}$, $a_i \in d(i, p_j)$. Let $out(s, F_\Gamma)$ be the set of all F_Γ-compatible paths from s.

We can now assign a meaning to ATL^* formulas on iCGS based on a semantics with two truth values: ff and tt.

Definition 4 (Satisfaction)**.** *The two-valued satisfaction relation $\models^2$ for an iCGS M, state $s \in S$, path $p \in S^\omega$, atom $q \in AP$, and ATL^* formula ϕ is defined as follows (clauses for Boolean connectives are immediate and thus omitted):*

$(M,s) \models^2 q$ iff $V(s,q) = \mathrm{tt}$
$(M,s) \models^2 \langle\langle\Gamma\rangle\rangle\psi$ iff for some F_Γ, for all $p \in out(s, F_\Gamma)$, $(M,p) \models^2 \psi$
$(M,p) \models^2 \varphi$ iff $(M,p_1) \models^2 \varphi$
$(M,p) \models^2 X\psi$ iff $(M,p_{\geq 2}) \models^2 \psi$
$(M,p) \models^2 \psi U \psi'$ iff for some $k \geq 1$, $(M,p_{\geq k}) \models^2 \psi'$, and for all j, $1 \leq j < k \Rightarrow (M,p_{\geq j}) \models^2 \psi$

We say that formula φ is *true* in an iCGS M, or $M \models^2 \varphi$, iff $(M, s_0) \models^2 \varphi$.

We now state the model checking problem within the two-valued semantics.

Definition 5 (Model Checking)**.** *Given an iCGS M and a formula ϕ, the model checking problem concerns determining whether $M \models^2 \phi$.*

Since the semantics provided in Def. 4 is the standard interpretation of ATL^* (Alur, Henzinger, and Kupferman 2002; Jamroga and van der Hoek 2004), it is well known that model checking ATL, *a fortiori* ATL^*, against iCGS with imperfect information and perfect recall is undecidable (Dima and Tiplea 2011). In the rest of the paper we develop methods to obtain partial solutions to this; but first we illustrate the formal machine above with a toy example.

Example 1. *The iCGS M depicted in Fig. 1 describes a variant of the Train Gate Controller scenario (Alur, Henzinger, and Kupferman 2002). Two trains t_1 and t_2 pass through a crossroad. Due to agreements between the railway companies, train t_1 can choose between the right (r) or left (l) track, while t_2 can choose between the right (r), left (l) or straight (s) track. At the same time, controller c has to select the right combination of tracks. For example, if t_1 and t_2 choose the joint action rs, then c has to select action 1 to proceed to the next step. Moreover, train t_1 has partial observability on the choices of t_2. For instance, if t_1 chooses l, then she cannot distinguish whether t_2 selects r or s, but she would observe if t_2 chose l as well.*

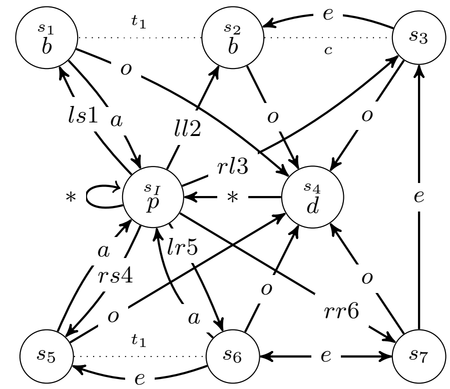

Figure 1: The iCGS M for Example 1. Notice that the transitions are generated with triples of actions. To improve readability every occurrence of the action idle (i) is omitted. Moreover, $*$ denotes any tuple of actions for which a transition is not given explicitly.

After this first step, c can still change her mind. Specifically, she can change arbitrarily the selection of tracks (e), request a new choice to the trains (a), or execute their selection (o). The controller c has partial observability, she cannot distinguish between s_2 and s_3, i.e. she does not distinguish r and l of t_1 when t_2 selects l. Finally, we use three atoms, one to denote the initial state (p), one for the preferred selections for t_1 (b), and one to mark that an agreement has been reached amongst the players (d). More formally, the iCGS M is comprised of the agents in $Ag = \{t_1, t_2, c\}$, atoms in $AP = \{p, b, d\}$, states in $S = \{s_I, s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$ with initial state s_I, actions in $Ac_{t_1} = \{r, l, i\}$, $Ac_{t_2} = \{r, l, s, i\}$, $Ac_c = \{1, 2, 3, 4, 5, 6, a, e, o, i\}$. Transitions are given as in Fig. 1, and we have the following indistinguishability between different states (indistinguishability is reflexive as well): $s_1 \sim_{t_1} s_2$, $s_5 \sim_{t_1} s_6$, and $s_2 \sim_c s_3$.

As an example of specifications in ATL^*, consider the formula $\varphi = \langle\langle\Gamma\rangle\rangle F(b \wedge \neg p U d)$, for $\Gamma = \{t_1, c\}$. This formula can be read as: controller c and train t_1 have a joint strategy such that eventually one of the preferred selections for t_1 is visited, and then an agreement has to be reached before visiting the initial state again. Notice that φ is true in M, while for $\Gamma = \{c\}$, it is false as, whenever t_1 always chooses r and t_2 always chooses s, then controller c cannot make b true before d holds. Finally, consider the ATL formula $\langle\langle Ag\rangle\rangle Fd$, whereby all agents aim at reaching an agreement, thus making the railway work, which can be seen to be true in M. However, given the undecidability of the corresponding model checking problem, there is no general method to verify specifications like these on any given iCGS. Hereafter we provide a sound, albeit partial, method to tackle this problem.

3 Three-valued Imperfect Information

In this section we introduce a novel generalisation of iCGS in terms of over- and under-approximations. Then, we develop a three-valued semantics for ATL^*, and show that it conservatively extends the two-valued semantics of the previous section. In what follows, for $x = may$ (resp. $must$), $\overline{x} = must$ (resp. may).

Definition 6 (Generalized iCGS). *Given sets Ag of agents and AP of atoms, a* generalized iCGS (with imperfect information) *is a tuple $M = \langle Ag, AP, S, s_0, \{Act_i\}_{i \in Ag}, d^{may}, d^{must}, \delta^{may}, \delta^{must}, \{\sim_i\}_{i \in Ag}, V \rangle$ such that:*

1. *S, s_0, $\{Act_i\}_{i \in Ag}$, $\{\sim_i\}_{i \in Ag}$ are defined as in Def. 1.*
2. *d^{may} and d^{must} are* protocol functions *from $Ag \times S$ to $2^{Act} \setminus \emptyset$ such that for every $i \in Ag$ and $s \in S$, (i) $d^{must}(i, s) \subseteq d^{may}(i, s) \subseteq Act_i$ and (ii) $s \sim_i s'$ implies $d^x(i, s) = d^x(i, s')$.*
3. *δ^{may} and δ^{must} are* transition relations *on $S \times ACT \times S$ such that $s' \in \delta^x(s, \vec{a})$ is defined for some $s' \in S$ only if $a_i \in d^x(i, s)$ for every $i \in Ag$. Moreover, $\delta^{must}(s, \vec{a}) \subseteq \delta^{may}(s, \vec{a})$.*
4. *$V : S \times AP \to \{\text{tt}, \text{ff}, \text{uu}\}$ is the* three-valued labelling function.

Intuitively, $must$-components are more restrictive than may-components: $must$-transitions can be interpreted as under-approximations of the actual transitions in the iCGS, while may-transitions can be thought of as over-approximations. The undefined value uu can be interpreted in various ways, for instance, unknown, unspecified, or inconsistent, depending on the application in hand. This is standard in multi-valued abstraction based methods (Shoham and Grumberg 2004; Ball and Kupferman 2006) and we do not discuss this further. We say that the truth value τ is *defined* whenever $\tau \neq$ uu. In the case that under- and over-approximations coincide, i.e., $d^{may} = d^{must}$ and $\delta^{may} = \delta^{must}$, and the truth value of every atom is defined, then we have a standard iCGS as per Def. 1. On the other hand, if each equivalence relation $\sim_i$ is the identity, then we have a generalized CGS (with perfect information).

Next, we introduce $must$- and may-strategies.

Definition 7 (Uniform x-Strategy with Perfect Recall). *For $x \in \{may, must\}$, a* uniform x-strategy with perfect recall *for agent $i \in Ag$ is a function $f_i^x : S^+ \to Act_i$ such that for every history $h, h' \in S^+$, (i) $f_i^x(h) \in d^x(i, last(h))$; and (ii) if $h \sim_i h'$ then $f_i^x(h) = f_i^x(h')$.*

Here we distinguish between may and $must$ strategies to over- and under-approximate the strategic abilities of agents. Again, the distinction collapses in the case of standard (two-valued) iCGS.

For $x \in \{may, must\}$ and a joint strategy $F_\Gamma^x = \{f_i^x \mid i \in \Gamma\}$, a path $p \in S^\omega$ is F_Γ^x*-compatible* iff for every $j \geq 1$, $p_{j+1} = \delta^{\overline{x}}(p_j, \vec{a})$ for some joint action $\vec{a}$ such that for every $i \in \Gamma$, $a_i = f_i^x(p_{\leq j})$, and for every $i \notin \Gamma$, $a_i \in d^{\overline{x}}(i, p_j)$. Then, let $out(s, F_\Gamma^x)$ be the set of all F_Γ^x-compatible paths starting from s. We report full definitions in Table 1.

Intuitively, when computing the outcomes of a joint strategy F_Γ^{must} from state s, we adopt a "conservative" stance with respect to the abilities of agents in Γ, by considering only actions enabled according to the under-approximated protocol d^{must}, as well as an "optimistic" stance about the capabilities of agents in $\overline{\Gamma}$, as given by the over-approximated protocol d^{may} and transition δ^{may}. For $out(s, F_\Gamma^{may})$ the reasoning is symmetric (notice that it might be empty in general). This modelling choice is in line with similar three-valued semantics for logics of strategies (Ball and Kupferman 2006; Lomuscio and Michaliszyn 2016).

Formally we define the three-valued semantics for ATL^* as follows.

Definition 8 (Satisfaction). *The three-valued satisfaction relation $\models^3$ for an iCGS M, state $s \in S$, path $p \in S^\omega$, atom $q \in AP$, $v \in \{\text{tt}, \text{ff}\}$, and ATL^* formula ϕ is defined as in Table 2. In all other cases the value of ϕ is* uu.

Observe that, in the clauses for ATL^* operators $must$-strategies are used to check the truth of formulas, while may-strategies appear in the clauses for falsehood. Specifically, to check whether $((M, s) \models^3 \langle\langle\Gamma\rangle\rangle\psi) = \text{tt}$ we consider all paths in $out(s, F_\Gamma^{must})$, which are defined by δ^{may}-transitions. This restricts the choices available to coalition Γ, while increasing the number of paths in which the formula needs to be satisfied. Similarly, to verify whether $((M, s) \models^3 \langle\langle\Gamma\rangle\rangle\psi) = \text{ff}$ we need to use δ^{must}-transitions over the paths in $out(s, F_\Gamma^{may})$, so as to reduce the number of candidates witnessing the falsehood of the formula. Notice also that, as regards Boolean operators, our semantics correspond to Kleene's three-valued logic.

Finally, $(M \models^3 \varphi) = \text{tt}$ (resp. ff) iff $((M, s_0) \models^3 \varphi) = \text{tt}$ (resp. ff). Otherwise, $(M \models^3 \varphi) = \text{uu}$.

We conclude this section by proving some auxiliary results on conservative extensions and the model checking problem.

Lemma 1 (Conservativeness). *Let M be a standard iCGS, that is, $d^{may} = d^{must}$, $\delta^{may} = \delta^{must}$ are functions, and the truth value of every atom is defined. Then, for every formula ϕ in ATL^*,*

$$((M, s) \models^3 \phi) = \text{tt} \quad \Leftrightarrow \quad (M, s) \models^2 \phi \tag{1}$$

$$((M, s) \models^3 \phi) = \text{ff} \quad \Leftrightarrow \quad (M, s) \not\models^2 \phi \tag{2}$$

By Lemma 1 the three-valued semantics for ATL^* is a conservative extension of its two-valued semantics, as the two coincide whenever we consider standard iCGS. Thus, from the results in the previous section it immediately follows that model checking ATL^* formulas under the three-valued semantics, with imperfect information and perfect recall is also undecidable. However, for perfect information we can show the following.

Theorem 1. *The model checking problem for generalized CGS (with perfect information) is 2EXPTIME-complete for ATL^* and PTIME-complete for ATL.*

In the following section we leverage on the decidable model checking problem for the three-valued semantics under perfect information to develop a sound, albeit incomplete, abstraction-based method to verify imperfect information.

$$out(s, F_\Gamma^{must}) = \{p \in S^\omega \mid \text{ for all } j \geq 0, p_{j+1} \in \delta^{may}(p_j, (F_\Gamma^{must}(p_{\leq j}), \vec{a}_{\overline{\Gamma}})) \text{ and for all } i \in \overline{\Gamma}, a_i \in d^{may}(i, p_j)\}$$
$$out(s, F_\Gamma^{may}) = \{p \in S^\omega \mid \text{ for all } j \geq 0, p_{j+1} \in \delta^{must}(p_j, (F_\Gamma^{may}(p_{\leq j}), \vec{a}_{\overline{\Gamma}})) \text{ and for all } i \in \overline{\Gamma}, a_i \in d^{must}(i, p_j)\}$$

Table 1: The definitions of $out(s, F_\Gamma^{must})$ and $out(s, F_\Gamma^{may})$.

$((M,s) \models^3 q) = v$	iff	$V(s,q) = v$
$((M,s) \models^3 \neg\varphi) = v$	iff	$((M,s) \models^3 \varphi) = \neg v$
$((M,s) \models^3 \varphi \wedge \varphi') = \mathrm{tt}$	iff	$((M,s) \models^3 \varphi) = \mathrm{tt}$ and $((M,s) \models^3 \varphi') = \mathrm{tt}$
$((M,s) \models^3 \varphi \wedge \varphi') = \mathrm{ff}$	iff	$((M,s) \models^3 \varphi) = \mathrm{ff}$ or $((M,s) \models^3 \varphi') = \mathrm{ff}$
$((M,s) \models^3 \langle\langle\Gamma\rangle\rangle\psi) = \mathrm{tt}$	iff	for some F_Γ^{must}, for all $p \in out(s, F_\Gamma^{must})$, $((M,p) \models^3 \psi) = \mathrm{tt}$
$((M,s) \models^3 \langle\langle\Gamma\rangle\rangle\psi) = \mathrm{ff}$	iff	for every F_Γ^{may}, for some $p \in out(s, F_\Gamma^{may})$, $((M,p) \models^3 \psi) = \mathrm{ff}$
$((M,p) \models^3 \varphi) = v$	iff	$((M,p_1) \models^3 \varphi) = v$
$((M,p) \models^3 \neg\psi) = v$	iff	$((M,p) \models^3 \psi) = \neg v$
$((M,p) \models^3 \psi \wedge \psi') = \mathrm{tt}$	iff	$((M,p) \models^3 \psi) = \mathrm{tt}$ and $((M,p) \models^3 \psi') = \mathrm{tt}$
$((M,p) \models^3 \psi \wedge \psi') = \mathrm{ff}$	iff	$((M,p) \models^3 \psi) = \mathrm{ff}$ or $((M,p) \models^3 \psi') = \mathrm{ff}$
$((M,p) \models^3 X\psi) = v$	iff	$((M,p_{\geq 2}) \models^3 \psi) = v$
$((M,p) \models^3 \psi U \psi') = \mathrm{tt}$	iff	for some $k \geq 1$, $((M,p_{\geq k}) \models^3 \psi') = \mathrm{tt}$, and for all j, $1 \leq j < k \Rightarrow ((M,p_{\geq j}) \models^3 \psi) = \mathrm{tt}$
$((M,p) \models^3 \psi U \psi') = \mathrm{ff}$	iff	for all $k \geq 1$, $((M,p_{\geq k}) \models^3 \psi') = \mathrm{ff}$, or for some $j \geq 1$, $((M,p_{\geq j}) \models^3 \psi) = \mathrm{ff}$, and for all j', $1 \leq j' \leq j \Rightarrow ((M,p_{\geq j'}) \models^3 \psi') = \mathrm{ff}$

Table 2: The three-valued satisfaction relation for ATL^*.

4 Abstraction

We now define perfect information, three-valued abstractions for iCGS. Then, we show that defined truth values for ATL^* formulas transfer from such abstractions to the original iCGS with imperfect information. Since the model checking problem on the former is decidable (as per Theorem 1), this preservation result can be used to define a sound, albeit partial, verification procedure under imperfect information and perfect recall.

To begin with, given a coalition $\Gamma \subseteq Ag$ of agents, define the *common knowledge relation* $\sim_\Gamma^C$ as the reflexive and transitive closure $(\bigcup_{i\in\Gamma} \sim_i)^*$ of the union of indistinguishability relations $\sim_i$ for $i \in \Gamma$ (Fagin et al. 1995). That is, $s \sim_\Gamma^C s'$ iff s' is reachable from s by a sequence $s_1, \ldots, s_n$ of states such that (i) $s_1 = s$, (ii) $s_n = s'$, and (iii) for every $j < n$, $s_j \sim_i s_{j+1}$ for some $i \in \Gamma$. Clearly, $\sim_\Gamma^C$ is an equivalence relation. Now, let $[s]_\Gamma = \{s' \in S \mid s' \sim_\Gamma s\}$ be the equivalence class of s according to $\sim_\Gamma$. The relation $\sim_\Gamma$ is extended to histories in a synchronous, pointwise way, i.e., given $h, h' \in S^+$, $h \sim_\Gamma h'$ iff (i) $|h| = |h'|$ and (ii) for all $j \leq |h|$, $h_j \sim_\Gamma h'_j$. So, we introduce the notation $[h]_\Gamma = \{h' \in S^+ \mid h' \sim_\Gamma h\}$.

Now, we introduce abstractions for iCGS.

Definition 9 (Abstract CGS). *Given an iCGS* $M = \langle Ag, AP, S, s_0, \{Act_i\}_{i\in Ag}, d, \delta, \{\sim_i\}_{i\in Ag}, V\rangle$ *and a coalition* $\Gamma \subseteq Ag$, *the* abstract (generalized) CGS $M_\Gamma = \langle Ag, AP, S_\Gamma, [s_0]_\Gamma, \{Act_i\}_{i\in Ag}, d_\Gamma^{may}, d_\Gamma^{must}, \delta_\Gamma^{may}, \delta_\Gamma^{must}, V_\Gamma\rangle$ *is defined such that:*

1. *$S_\Gamma = \{[s]_\Gamma \mid s \in S\}$ is the set of equivalence classes for all states $s \in S$, with initial state $[s_0]_\Gamma$;*
2. *for every $t, t' \in S_\Gamma$ and joint action $\vec{a}$, $t' \in \delta_\Gamma^{may}(t, \vec{a})$ iff for some $s \in t$ and $s' \in t'$, $\delta(s, \vec{a}) = s'$;*
3. *for every $t, t' \in S_\Gamma$ and joint action $\vec{a}$, $t' \in \delta_\Gamma^{must}(t, \vec{a})$ iff for all $s \in t$ there is $s' \in t'$ such that $\delta(s, \vec{a}) = s'$;*
4. *for $x \in \{may, must\}$, $t \in S_\Gamma$, and $i \in Ag$, $d_\Gamma^x(i,t) = \{a_i \in Act_i \mid \delta_\Gamma^x(t, (a_i, \vec{a}_{\bar{i}}))$ is defined for some $\vec{a}_{\bar{i}}\}$;*
5. *for $v \in \{\mathrm{tt}, \mathrm{ff}\}$, $p \in AP$, and $t \in S_\Gamma$, $V_\Gamma(t,p) = v$ iff $V(s,p) = v$ for all $s \in t$; otherwise, $V_\Gamma(t,p) = \mathrm{uu}$.*

We now show that the abstraction of an iCGS is indeed a generalized CGS (with perfect information) as defined in Def. 6. In particular, the indistinguishability relation for every $i \in Ag$ is assumed to be the identity relation.

Lemma 2. *For every coalition $\Gamma \subseteq Ag$, any abstraction M_Γ of an iCGS M is a generalized CGS.*

We can now state the main theoretical result in this section, namely if a Γ-formula has a defined truth value in an abstract CGS M_Γ, built on an iCGS M, then the Γ-formula has the same truth value in M.

Theorem 2. *Given an iCGS M, state s, and coalition $\Gamma \subseteq Ag$, for every Γ-formula ϕ in ATL^*, we have that*

$$((M_\Gamma, [s]_\Gamma) \models^3 \phi) = \mathrm{tt} \quad \Rightarrow \quad (M,s) \models^2 \phi \qquad (3)$$
$$((M_\Gamma, [s]_\Gamma) \models^3 \phi) = \mathrm{ff} \quad \Rightarrow \quad (M,s) \not\models^2 \phi \qquad (4)$$

By Theorem 2 a defined answer to the model checking problem w.r.t. abstract, generalized CGS (with perfect information), which is decidable, can be tranferred to the concrete, two-valued iCGS (with imperfect information), whose model checking problem is undecidable in general. Obviously, if the returned value is undefined (uu), then no conclusive answer can be drawn.

We illustrate the abstraction procedure with our Train Gate Controller scenario in Example 1.

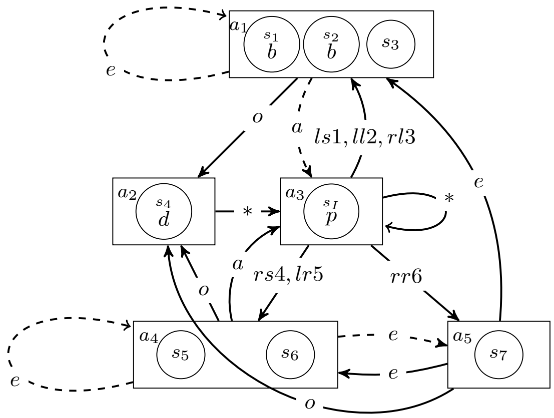

Figure 2: The abstract CGS for the iCGS in Example 1, where $must$-transitions are depicted with continuous lines while may-transitions are both the continuous and dashed lines.

Example 2. *In Fig. 2 we show the abstract CGS obtained from the iCGS for the Train Gate Controller scenario in Example 1 by considering the formula* $\varphi = \langle\langle\Gamma\rangle\rangle F(b \wedge \neg pUd)$ *for* $\Gamma = \{t_1, c\}$. *Specifically, the abstraction* M_Γ *includes five abstract states according to the equivalence relation* $\sim^C_{\{t_1,c\}}$. *Notice that formula* φ *is undefined in* M_Γ *due to the undefined value of atom* b *in the abstract state* a_1.

5 Refinement

By Theorem 2 if a formula is undefined on abstraction M_Γ, then no conclusion can be drawn on the model checking problem for M. In this section we provide a refinement procedure taking as input a "failure" state s_f in M_Γ and a Γ-formula φ such that φ is undefined in s_f, and returning a refined CGS M^r_Γ, whose state space is smaller than M in general, and for which we are able to prove Theorem 3, a preservation result similar to Theorem 2. In what follows we assume that failure states are identified manually. We leave their automatic generation for further work.

The algorithm $Refinement(M_\Gamma, M, s_f)$ is described in Fig. 3a. Intuitively, we look at incoming transitions into s_f. For concrete states s and s' in s_f, if the Γ-component of actions ending respectively in s and s' are different, any uniform strategy for Γ will visit either s or s'. As a result, the abstract state s_f can be split "safely" into an s- and an s'-component. More precisely, the procedure $Refinement()$ begins by initializing as true the values of a matrix m that stores the relation outlined above between the concrete states in s_f (line 1). Then, the algorithm calls the subroutine $Check_1(M_\Gamma, M, s_f, m)$ in Fig. 3b, which updates the values in m by considering the concrete transition function δ in M. In particular, at each iteration $Check_1()$ considers one predecessor t_f of s_f (line 1). Then, two other loops consider pairs of states s and s' in the abstract state s_f and pairs of states t and t' in the predecessor t_f (lines 2-3). If s and s' are indistinguishable for some agent $i \in \Gamma$ and i performs the same action in the transitions from t and t' to s and s' respectively (lines 4-6), then we update the value of the corresponding cell in m to false (line 6). The subroutine reported in $Check_1()$ carries out the first round of updates on m. Further updates in the $Refinement()$ algorithm are performed by the subroutine $Check_2(M_\Gamma, s_f, m, update)$ reported in Fig. 3c, which considers the "indirect" binding that some concrete states may have in an abstract state. Specifically, given the states s and s' in the abstract state s_f that have $true$ as value in m (lines 2-3), we need to consider the relation that s and s' have with the other states in s_f (lines 4-6): if the values in m for both states related with some other state t are $false$, then we update the value of cell $m[s, s']$ to $false$ as well. Subroutine $Check_2()$ is called repeatedly in algorithm $Refinement()$ as long as guard $update$ remains $true$. When $update$ becomes $false$, we proceed to check whether there is at least an element $true$ in m (line 8). If this is the case, we assign the related concrete states s and s' to two different, new abstract states v and w (line 10). Finally, we populate the new abstract states v and w with the other concrete states in the old abstract state s_f (which is removed) according to matrix m (lines 12-14).

Hereafter we present the formal definition of the refined CGS M^r_Γ as obtained by the application of the $Refinement()$ algorithm.

Definition 10 (Refined CGS)**.** *Given an abstract CGS* $M_\Gamma = \langle Ag, AP, S_\Gamma, s_0, \{Act_i\}_{i\in Ag}, d^{may}_\Gamma, d^{must}_\Gamma, \delta^{may}_\Gamma, \delta^{must}_\Gamma, V_\Gamma\rangle$, *its refinement* $M^r_\Gamma = \langle Ag, AP, S^r_\Gamma, s^r_0, \{Act_i\}_{i\in Ag}, d^{may}_\Gamma, d^{must}_\Gamma, \delta^{may}_\Gamma, \delta^{must}_\Gamma, V^r_\Gamma\rangle$ *as obtained by an application of algorithm* $Refinement(M_\Gamma, M, s_f)$ *is defined as follows:*

1. S^r_Γ *is the set* S_Γ *of states in* M_Γ*, possibly without the "failure" state* s_f*, but with the new states added by* $Refinement()$*. Then,* s^r_0 *is the state in* S^r_Γ *such that* $s_0 \in s^r_0$*, for* $s_0 \in M$.
2. *For* $x \in \{may, must\}$*, the transitions relations* δ^x_Γ *and the protocol functions* d^x_Γ *are defined as in Def. 9. In particular,*
   (a) *for every* $t, t' \in S^r_\Gamma$ *and joint action* $\vec{a}$, $t' \in \delta^{may}_\Gamma(t, \vec{a})$ *iff for some* $s \in t$ *and* $s' \in t'$, $\delta(s, \vec{a}) = s'$*;*
   (b) *for every* $t, t' \in S^r_\Gamma$ *and joint action* $\vec{a}$, $t' \in \delta^{must}_\Gamma(t, \vec{a})$ *iff for all* $s \in t$ *there is* $s' \in t'$ *such that* $\delta(s, \vec{a}) = s'$*;*
   (c) *for every* $t \in S^r_\Gamma$*, and* $i \in Ag$, $d^x_\Gamma(i, t) = \{a_i \in Act_i \mid \delta^x_\Gamma(t, (a_i, \vec{a}_{\bar{i}}))$ *is defined for some* $\vec{a}_{\bar{i}}\}$.
3. *For* $v \in \{\text{tt}, \text{ff}\}$, $p \in AP$*, and* $t \in S^r_\Gamma$, $V^r_\Gamma(t, p) = v$ *iff* $V(s, p) = v$ *for all* $s \in t$*; otherwise,* $V_\Gamma(s, p) = \text{uu}$.

By Def. 10 the components of the refined CGS M^r_Γ coincide with those in abstraction M_Γ, except possibly as regards the "failure" state s_f and new states introduced by $Refinement()$. On the new states, the transition relations and protocol functions are defined in analogy with M_Γ.

We now show a property of the refined CGS M^r_Γ, which will be useful to prove the main preservation result Theorem 3. Intuitively, must strategies in M^r_Γ respect uniformity *on the set of their outcomes*.

Algorithm $Refinement(M_\Gamma, M, s_f)$:

for $s, s' \in s_f, m[s,s'] = true$;
$Check_1(M_\Gamma, M, s_f, m)$;
$update = true$;
while $update = true$
$Check_2(M_\Gamma, s_f, m, update)$;
$split = false$;
while $s, s' \in s_f$ **and** $split = false$
if $m[s,s'] = true$ **then**
$remove(s_f, S_\Gamma)$;
$add(v, S_\Gamma); add(w, S_\Gamma); add(s, v); add(s', w)$;
$split = true$;
for $t \in s_f$
if $m[s,t] = true$ **then** $add(t, w)$;
else $add(t, v)$;

(a)

Algorithm $Check_1(M_\Gamma, M, s_f, m)$:

for $t_f \in Pre(s_f)$
for $s, s' \in s_f$
for $t, t' \in t_f$
if $\delta(t, \vec{a}) = s$ **and** $\delta(t', \vec{b}) = s'$ **then**
for $i \in \Gamma$
if $s \sim_i s'$ **and** $\vec{a}_i = \vec{b}_i$ **then** $m[s,s'] = false$;

(b)

Algorithm $Check_2(M_\Gamma, s_f, m, update)$:

$update = false$;
for $s, s' \in s_f$
if $m[s,s'] = true$ **then**
for $t \in s_f$
if $m[s,t] = false$ **and** $m[s',t] = false$ **then**
$m[s,s'] = false$;
$update = true$;

(c)

Figure 3: The *Refinement procedure (3a) with its auxiliary subroutines Check*$_1$ *and Check*$_2$ *(3b and 3c respectively).*

Lemma 3. *In* M_Γ^r *for every joint strategy* F_Γ^{must}*, for all* $p, \hat{p} \in out(t, F_\Gamma^{must})$*, all* $p' \in p$*,* $\hat{p}' \in \hat{p}$*, and all* $i \in \Gamma$*,* $j \in \mathbb{N}$*, if* $p'_{\leq j} \sim_i \hat{p}'_{\leq j}$ *then* $f_i^{must}(p_{\leq j}) = f_i^{must}(\hat{p}_{\leq j})$*.*

By Lemma 3 we can prove the main preservation result of this section. In particular, the lemma is used in the inductive step for strategy operators.

Theorem 3. *Given an iCGS* M*, state* s*, coalition* Γ*, its abstract CGS* M_Γ *with refinement* M_Γ^r*, and state* $s_\Gamma^r \ni s$*, for every* Γ*-formula* ϕ *in* ATL^**,*

$$((M_\Gamma^r, s_\Gamma^r) \models^3 \phi) = \text{tt} \quad \Rightarrow \quad (M, s) \models^2 \phi \qquad (5)$$

$$((M_\Gamma^r, s_\Gamma^r) \models^3 \phi) = \text{ff} \quad \Rightarrow \quad (M, s) \not\models^2 \phi \qquad (6)$$

By Theorem 3 defined truth values are preserved from the refined CGS to the original iCGS, similarly to Theorem 2.

Example 3. *In Fig. 4 we present a refinement of the abstract CGS in Fig. 2. In this new model we split the state* a_1 *in two new abstract states* a_1^1 *and* a_1^2 *according to the* $Refinement()$ *algorithm in Fig. 3a. By doing so, formula* $\varphi = \langle\langle\{t_1, c\}\rangle\rangle F(b \wedge \neg pUd)$ *becomes true as atom* b *becomes defined in* a_1^1 *and* a_1^2*. So, by Theorem 3 formula* φ *is true in the original iCGS* M *as well.*

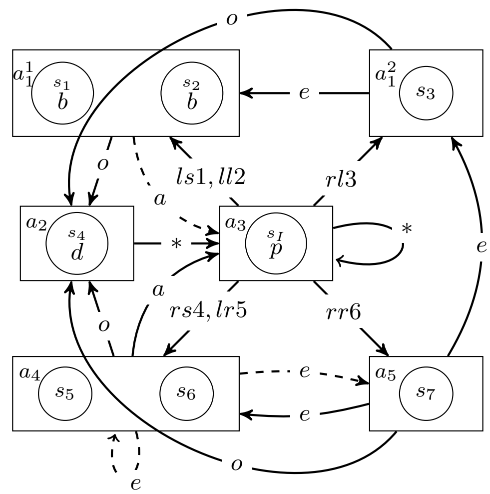

Figure 4: Example of split of abstract CGS in Fig. 2.

By combining the results in Section 3, 4, and 5 we can outline a method to verify strategic properties of Multi-Agent Systems under the assumptions of imperfect information and perfect recall. Given an iCGS M and a Γ-formula ϕ in ATL^*, we first build the abstract, three-valued CGS M_Γ as per Def. 9. We can model check ϕ on M_Γ, as the corresponding decision problem is decidable by Theorem 1, and then tranfer any defined answer to the original iCGS M in virtue of Theorem 2. In case of an undefined answer, we can apply the refinement procedure in Section 5 iteratively: if the value of ϕ or any of its subformulas is undefined at some state s_f in M_Γ, we can apply the refinement algorithm so as to obtain a refined CGS M_Γ^r: any defined value for ϕ on M_Γ^r transfers to M by Theorem 3. The refinement step can be iterated as long as ϕ stays undefined. Since the verification of ATL^* under imperfect information and perfect recall is undecidable in general (Dima and Tiplea 2011), the procedure here outlined is obviously partial and there is no guarantee of termination with a defined answer. However, partial results can be useful in cases of interest, like the Train Gate Controller scenario illustrated in Example 1, 2, and 3.

6 Conclusions

As we discussed in the introduction one of the key issues in employing logics for strategic reasoning, such as ATL and ATL^*, in the context of Multi-agent Systems is that their model checking problem is undecidable under perfect recall and incomplete information. Yet, this is one of the most natural and compelling setup in applications. Finding appropriate approximations remains an open problem at present.

In this paper we have put forward a notion of abstraction between different classes of systems to overcome this difficulty. Specifically, we showed that iCGS with imperfect information admit a (perfect information) abstraction which preserves satisfaction back to the original model, when checked under a three-valued semantics. This enabled us to give an incomplete but sound procedure for the original model checking problem, which is undecidable in general.

In future work we intend to build a toolkit to generate abstractions and refinements automatically, perhaps in combination with refinement techniques built on interpolants (Ball and Kupferman 2006). Moreover, we plan to extend the abstraction and refinement techniques here developed to more expressive languages for strategic reasoning including Strategy Logic (Chatterjee, Henzinger, and Piterman 2007; Mogavero et al. 2014).

Acknowledgements. F. Belardinelli acknowledges the support of ANR JCJC Project SVeDaS (ANR-16-CE40-0021). A. Lomuscio is supported by a Royal Academy of Engineering Chair in Emerging Technologies.

References

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

Aminof, B.; Kupferman, O.; and Murano, A. 2012. Improved model checking of hierarchical systems. *Inf. Comput.* 210:68–86.

Avni, G.; Guha, S.; and Kupferman, O. 2017. An abstraction-refinement methodology for reasoning about network games. In IJCAI'17, 70–76.

Baier, C., and Katoen, J. P. 2008. *Principles of Model Checking (Representation and Mind Series)*.

Ball, T., and Kupferman, O. 2006. An abstraction-refinement framework for multi-agent systems. In LICS'06, 379–388.

Belardinelli, F., and Lomuscio, A. 2016. A three-value abstraction technique for the verification of epistemic properties in multi-agent systems. In JELIA'16, 112–126.

Belardinelli, F.; Lomuscio, A.; Murano, A.; and Rubin, S. 2017a. Verification of broadcasting multi-agent systems against an epistemic strategy logic. In IJCAI'17, 91–97.

Belardinelli, F.; Lomuscio, A.; Murano, A.; and Rubin, S. 2017b. Verification of multi-agent systems with imperfect information and public actions. In AAMAS'17, 1268–1276.

Belardinelli, F.; Lomuscio, A.; and Malvone, V. 2018. Approximating perfect recall when model checking strategic abilities. In KR'18, 435–444.

Berthon, R.; Maubert, B.; Murano, A.; Rubin, S.; and Vardi, M. Y. 2017. Strategy logic with imperfect information. In LICS'17, 1–12.

Chatterjee, K.; Henzinger, T.; and Piterman, N. 2007. Strategy logic. In CONCUR'07, volume 4703, 59–73.

Cohen, M.; Dam, M.; Lomuscio, A.; and Russo, F. 2009. Abstraction in model checking multi-agent systems. In AAMAS'09, 945–952.

de Alfaro, L., and Roy, P. 2010. Solving games via three-valued abstraction refinement. *Inf. Comput.* 208(6):666–676.

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

Dimitrova, R., and Finkbeiner, B. 2008. Abstraction refinement for games with incomplete information. In FSTTCS'08, 175–186.

Fagin, R.; Halpern, J.; Moses, Y.; and Vardi, M. 1995. *Reasoning about Knowledge.*

Grumberg, O.; Lange, M.; Leucker, M.; and Shoham, S. 2007. When not losing is better than winning: Abstraction and refinement for the full mu-calculus. *Inf. Comput.* 205(8):1130–1148.

Jamroga, W., and van der Hoek, W. 2004. Agents that know how to play. *Fund. Inf.* 62:1–35.

Jamroga, W.; Konikowska, B.; and Penczek, W. 2016. Multi-valued verification of strategic ability. In AAMAS'16, 1180–1189.

Lomuscio, A., and Michaliszyn, J. 2014. An abstraction technique for the verification of multi-agent systems against ATL specifications. In KR'14, 428–437.

Lomuscio, A., and Michaliszyn, J. 2015. Verifying multi-agent systems by model checking three-valued abstractions. In AAMAS'15, 189–198.

Lomuscio, A., and Michaliszyn, J. 2016. Verification of multi-agent systems via predicate abstraction against ATLK specifications. In AAMAS'16, 662–670.

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

Raimondi, F., and Lomuscio, A. 2005. The complexity of symbolic model checking temporal-epistemic logics. In CS&P, 421–432.

Shoham, S., and Grumberg, O. 2004. Monotonic abstraction-refinement for CTL. In TACAS'04, 546–560.

Shoham, S., and Grumberg, O. 2007. A game-based framework for CTL counterexamples and 3-valued abstraction-refinement. *ACM Trans. Comp. Log.* 9(1):1.