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 behaviors 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 successive 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 $\mu$-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 games but they also refer at the 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 γi is the identity relation, i.e., s 1→ i s' iff s = s', we obtain a standard CGS with perfect information (Alur, Henzinger, and Kupferman 2000). Hereafter we consider both the class icGS of all iCGS, and its subclass CGS of all CGS with perfect information.

Given a set Γ ⊆ Ag of agents and a joint action a ∈ ACT, let aγ and aΓ be two tuples comprising only of actions for the agents in Γ, resp. Ω. We also write ai and aγ for a{1}i respectively. Finally, for a and b in ACT, (aγ, bγ) denotes the joint action where the actions for the agents in Γ (resp. Ω) are taken from a (resp. b).

A history h ∈ 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′ ∈ S+ are indistinguishable for agent i ∈ Ag, or h 1→ i h′, iff (i) |h| = |h′| and (ii) for all j ≤ |h|, hj 1→ i hj.

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 (ψ) and path (ψ) formulas in ATL* are defined as follows, where q ∈ AP and Γ ⊆ Ag:

\[
\begin{align*}
\varphi & ::= \quad q \mid \neg \varphi \mid \varphi \land \varphi \mid \langle \Gamma \rangle \psi \\
\psi & ::= \quad \varphi \mid \neg \psi \mid \psi \land \psi \mid X \psi \mid (\psi U \psi)
\end{align*}
\]

Formulas in ATL* are all and only the state formulas. As customary, a formula ⟨Γ⟩ϕ 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 [Γ], 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:

\[
\psi ::= \quad X \varphi \mid (\varphi U \varphi) \mid (\varphi R \varphi)
\]

Hereafter we also consider the fragment of Γ-formulas, i.e., formulas in which the strategic operator [Γ] ranges only over some coalition Γ ⊆ 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 $f_i : S^+ \rightarrow Act_i$, 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 $f_i : S^+ \rightarrow Act_i$.

Given an iCGS $M$, a path $p \in S^*$ is an infinite sequence $s_1 s_2 \ldots$ of states. Given a joint strategy $F_1 = \{ f_i \mid i \in \Gamma \}$, comprising of one strategy for each agent in coalition $\Gamma$, a path $p$ is $F_1$-compatible iff for every $j \geq 1, p_{j+1} = \delta(p_j, \bar{a})$ for some joint action $\bar{a}$ such that for every $i \in \Gamma, a_i = f_i(p_{\leq j})$, and for every $i \in \Gamma, a_i = d(i, p_{j})$. Let $\text{out}(s, F_1)$ be the set of all $F_1$-compatible paths from $s$.

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

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

$\models^2 (M,s) q \text{ iff } V(s,q) = tt$

$\models^2 (M,s) (\bigwedge \Gamma) \psi \text{ iff for some } F_1, \forall p \in \text{out}(s,F_1), (M,p) \models^2 \psi$

$\models^2 (M,p) \psi \text{ iff } (M,p_1) \models^2 \psi$

$\models^2 (M,p) X \psi \text{ iff } (M, p_{\geq 2}) \models^2 \psi$

$\models^2 (M,p) \psi U \psi' \text{ iff for some } k \geq 1, (M, p_{\geq k}) \models^2 \psi'$, and

for all $j, 1 \leq j < k \Rightarrow (M, p_{\geq j}) \models^2 \psi$

We say that formula $\varphi$ 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 $\varphi$, the model checking problem concerns determining whether $M \models^2 \varphi$.

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 $ATL, a fortiiori$ $ATL^*$, 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 road. 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.

After this first step, $c$ can still change her mind. Specifically, she can change arbitrarily the selection of tracks ($c$), 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 $S = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$ with initial state $s_1$, actions in $AC_{t_1} = \{r, l, i\}$, $AC_{t_2} = \{r, l, s, i\}$, $AC_c = \{1, 2, 3, 4, 5, 6, a, c, o, i\}$. Transitions are given as in Fig. 1, and we have the following indistinguishability between different states (indistinguishability is reflexive as well): $s_1 \sim_{t_1} s_2, s_2 \sim_{t_1} s_6$, and $s_2 \sim_{t_1} s_3$.

As an example of specifications in $ATL^*$, consider the formula $\varphi = \langle \Gamma \rangle F(b \land \neg p U d)$, 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 $\varphi$ 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 $\langle \text{Ag} \rangle Fd$, 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 $\text{ATL}^*$, and show that it conservatively extends the two-valued semantics of the previous section. In what follows, for $x = \text{may}$ (resp. $\text{must}$), $\models = \text{must}$ (resp. $\text{may}$).

Definition 6 (Generalized iCGS). Given sets $\mathcal{A}g$ of agents and $\mathcal{A}P$ of atoms, a generalized iCGS (with imperfect information) is a tuple $M = \langle \mathcal{A}g, \mathcal{A}P, \mathcal{S}0, \{\text{Act}_i\}_{i \in \mathcal{A}g}, d^{\text{may}}, d^{\text{must}}, \delta^{\text{may}}, \delta^{\text{must}}, \{\sim_i\}_{i \in \mathcal{A}g}, V \rangle$ such that:

1. $\mathcal{S}, \mathcal{S}0, \{\text{Act}_i\}_{i \in \mathcal{A}g}, \{\sim_i\}_{i \in \mathcal{A}g}$ are defined as in Def 1.
2. $d^{\text{may}}$ and $d^{\text{must}}$ are protocol functions from $\mathcal{A}g \times \mathcal{A}P$ to $2^{\mathcal{A}c} \setminus \emptyset$ such that for every $i \in \mathcal{A}g$ and $s \in \mathcal{S}$, (i) $d^{\text{must}}(i, s) \subseteq d^{\text{may}}(i, s) \subseteq \text{Act}_i$ and (ii) $s \sim_i s'$ implies $d^2(i, s) = d^2(i, s')$.
3. $\delta^{\text{may}}$ and $\delta^{\text{must}}$ are transition relations on $\mathcal{S} \times \mathcal{A}c \times \mathcal{S}$ such that $s' \in \delta^\prime(s, \vec{a})$ is defined for some $s' \in \mathcal{S}$ only if $a_i \in d^2(i, s)$ for every $i \in \mathcal{A}g$. Moreover, $\delta^{\text{must}}(s, \vec{a}) \subseteq \delta^{\text{may}}(s, \vec{a})$.
4. $V: \mathcal{S} \times \mathcal{A}c \rightarrow \{\text{tt}, \text{ff}, \text{uu}\}$ is the three-valued labelling function.

Intuitively, $\text{must}$-components are more restrictive than $\text{may}$-components: $\text{must}$-transitions can be interpreted as under-approximations of the actual transitions in the iCGS, while $\text{may}$-transitions can be thought of as over-approximations. The undefined value $\text{uu}$ 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 $\tau$ is defined whenever $\tau \neq \text{uu}$. In the case that under- and over-approximations coincide, i.e., $d^{\text{may}} = d^{\text{must}}$ and $\delta^{\text{may}} = \delta^{\text{must}}$, 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 $\text{must}$- and $\text{may}$-strategies.

Definition 7 (Uniform $x$-Strategy with Perfect Recall). For $x \in \{\text{may}, \text{must}\}$, a uniform $x$-strategy with perfect recall for agent $i \in \mathcal{A}g$ is a function $f^x_i: S^* \rightarrow \text{Act}_i$ such that for every history $h, h' \in S^*$, (i) $f^x_i(h) \in d^2(i, \text{last}(h))$; and (ii) if $h \sim_i h'$ then $f^x_i(h) = f^x_i(h')$.

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

For $x \in \{\text{may}, \text{must}\}$ and a joint strategy $F^x_t = \{f^x_i \mid i \in \Gamma\}$, a path $p \in S^*$ is $F^x_t$-compatible iff for every $j \geq 1$, $p_{j+1} = \delta^x(p_j, \vec{a})$ for some joint action $\vec{a}$ such that for every $i \in \Gamma$, $a_i = f^x_i(p_{j-1})$, and for every $i \in \Gamma$, $a_i \in d^2(i, p_j)$. Then, let $\text{out}(s, F^x_t)$ be the set of all $F^x_t$-compatible paths starting from $s$. We report full definitions in Table 1.

Intuitively, when computing the outcomes of a joint strategy $F_t^{\text{must}}$ from state $s$, we adopt a “conservative” stance with respect to the abilities of agents in $\Gamma$, by considering only actions enabled according to the under-approximated protocol $d^{\text{must}}$, as well as an “optimistic” stance about the capabilities of agents in $\overline{\Gamma}$, as given by the over-approximated protocol $d^{\text{may}}$ and transition $\delta^{\text{may}}$. For $\text{out}(s, F_t^{\text{may}})$ 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 $\text{ATL}^*$ as follows.

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

Observe that, in the clauses for $\text{ATL}^*$ operators $\text{must}$-strategies are used to check the truth of formulas, while $\text{may}$-strategies appear in the clauses for falsehood. Specifically, to check whether $\langle M, s \rangle \models^3 \langle \Gamma \rangle \psi$ we consider all paths in $\text{out}(s, F_{\psi}^{\text{must}})$, which are defined by $\delta^{\text{must}}$-transitions. This restricts the choices available to coalition $\Gamma$, while increasing the number of paths in which the formula needs to be satisfied. Similarly, to verify whether $\langle M, s \rangle \models^3 \langle \Gamma \rangle \psi$ we need to use $\delta^{\text{must}}$-transitions over the paths in $\text{out}(s, F_{\psi}^{\text{must}})$, 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, $(M, s) \models^3 \varphi = \text{tt}$ (resp. $\text{ff}$) iff $(M, s_0) \models^3 \varphi = \text{tt}$ (resp. $\text{ff}$). Otherwise, $(M, s) \models^3 \varphi = \text{uu}$.

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, $d^{\text{may}} = d^{\text{must}}, \delta^{\text{may}} = \delta^{\text{must}}$ are functions, and the truth value of every atom is defined. Then, for every formula $\phi$ in $\text{ATL}^*$,

\[
\langle M, s \rangle \models^3 \varphi = \text{tt} \iff \langle M, s \rangle \models^2 \varphi \quad (1)
\]

\[
\langle M, s \rangle \models^3 \varphi = \text{ff} \iff \langle M, s \rangle \models^2 \varphi \quad (2)
\]

By Lemma 1 the three-valued semantics for $\text{ATL}^*$ 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 $\text{ATL}^*$ 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 $2\text{EXPTIME}$-complete for $\text{ATL}^*$ and $\text{PTIME}$-complete for $\text{ATL}$. 

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_I\textsuperscript{\text{must}}) = \{p \in S' | \text{ for all } j \geq 0, p_{j+1} \in \delta^\text{mag}(p_j, (F_I\textsuperscript{\text{must}}(p_{j-1}), d\Gamma)) \text{ and for all } i \in \Gamma, a_i \in d^\text{mag}(i, p_j) \}

out(s, F_I\textsuperscript{\text{mag}}) = \{p \in S' | \text{ for all } j \geq 0, p_{j+1} \in \delta^\text{must}(p_j, (F_I\textsuperscript{\text{mag}}(p_{j-1}), d\Gamma)) \text{ and for all } i \in \Gamma, a_i \in d^\text{must}(i, p_j) \}

Table 1: The definitions of out(s, F_I\textsuperscript{\text{must}}) and out(s, F_I\textsuperscript{\text{mag}}).

| \begin{align*}
( (M, s) \models_3 q &= v  & \iff & (M, s) \models_3 \psi &= v \\
( (M, s) \models_3 \varphi \wedge \varphi' &= tt & \iff & ( (M, s) \models_3 \varphi &= tt \text{ and } ( (M, s) \models_3 \varphi' &= tt \\
( (M, s) \models_3 \varphi \wedge \varphi' &= ff & \iff & ( (M, s) \models_3 \varphi &= ff \text{ or } ( (M, s) \models_3 \varphi' &= ff \\
( (M, s) \models_3 (\Gamma \psi) &= tt & \iff & \text{ for some } F_I\textsuperscript{\text{must}}, \text{ for all } p \in out(s, F_I\textsuperscript{\text{must}}), ((M, p) \models_3 \psi) &= tt \\
( (M, s) \models_3 (\Gamma \psi) &= ff & \iff & \text{ for every } F_I\textsuperscript{\text{mag}}, \text{ for some } p \in out(s, F_I\textsuperscript{\text{mag}}), ((M, p) \models_3 \psi) &= ff \\
( (M, p) \models_3 \varphi &= v & \iff & ((M, p_1) \models_3 \varphi &= v \\
( (M, p) \models_3 \psi &= v & \iff & ((M, p) \models_3 \psi &= v \\
( (M, p) \models_3 \psi \wedge \psi &= tt & \iff & ((M, p) \models_3 \psi &= tt \text{ and } ((M, p) \models_3 \psi &= tt \\
( (M, p) \models_3 \psi &= ff & \iff & ((M, p) \models_3 \psi &= ff \text{ or } ((M, p) \models_3 \psi &= ff \\
( (M, p) \models_3 \psi &= v & \iff & ((M, p_2) \models_3 \psi &= v \\
( (M, p) \models_3 \psi &= v & \iff & \text{ for some } k \geq 1, ((M, p_{k+2}) \models_3 \psi &= tt \text{ and for all } j, 1 \leq j < k \Rightarrow ((M, p_{2j+1}) \models_3 \psi &= tt \\
( (M, p) \models_3 \psi &= v & \iff & \text{ for all } k \geq 1, ((M, p_{k+2}) \models_3 \psi &= ff, \text{ or for some } j \geq 1, ((M, p_{2j+1}) \models_3 \psi &= ff \text{ and for all } j', 1 \leq j' \leq j \Rightarrow ((M, p_{2j'+1}) \models_3 \psi &= ff \\
\end{align*} |

| 3. for every t, t' \in S_T and joint action \bar{a}, t' \in \delta^\text{must}(t, \bar{a}) if \ and for all s \in t there is s' \in t' such that δ(s, \bar{a}) = s' \\
| 4. for x \in \{\text{may, must}\}, t \in S_T, and i \in Ag, d^\text{I}(i, t) = \{a_i \in Act_i | \delta^\text{I}(t, (a_i, \bar{a}_i)) \text{ is defined for some } \bar{a}_i\} \\
| 5. for \bar{v} \in \{tt, ff\}, p \in AP, and t \in S_T, V_t(p, \bar{v}) = v if \ V(s, p) = v for all s \in t; otherwise, V_t(p, \bar{v}) = uu. |

We now show that the abstractions 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 \(M_T\) of an iCGS \(M\) is a generalized CGS.

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

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

\[ (M_T, [s]_\Gamma) \models_3 \phi = tt \Rightarrow (M, s) \models_2 \phi \]

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 transferred 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.
5 Refinement

By Theorem 2 if a formula is undefined on abstraction $M_f$, 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 $M_f$ and a formula $\varphi$ such that $\varphi$ is undefined in $s_f$, and returning a refined CGS $M'_f$, 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 $\text{Refinement}(M_f, M, s_f)$ 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 $\Gamma$-component of actions ending respectively in $s$ and $s'$ are different, any uniform strategy for $\Gamma$ 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 $\text{Refinement}()$ begins by initialising 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 $\text{Check}_1(M_f, M, s_f, m)$ in Fig. 3b, which updates the values in $m$ by considering the concrete transition function $\delta$ in $M$. In particular, at each iteration $\text{Check}_1()$ 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 $\text{Check}_1()$ carries out the first round of updates on $m$. Further updates in the $\text{Refinement}()$ algorithm are performed by the subroutine $\text{Check}_2(M_f, s_f, m, \text{update})$ 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 $\text{Check}_2()$ is called repeatedly in algorithm $\text{Refinement}()$ as long as guard $\text{update}$ remains $true$. When $\text{update}$ 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 $M'_f$ as obtained by the application of the $\text{Refinement}()$ algorithm.

**Definition 10 (Refined CGS).** Given an abstract CGS $M_f = \langle Ag, AP, S_f, s_0, \{Act_i\}_{i \in Ag}, d^{may}_f, d^{must}_f, \delta^{may}_f, \delta^{must}_f, V_f \rangle$, its refinement $M'_f = \langle Ag, AP, S'_f, s'_0, \{Act_i\}_{i \in Ag}, d^{may}_{r_f}, d^{must}_{r_f}, \delta^{may}_{r_f}, \delta^{must}_{r_f}, V'_f \rangle$ as obtained by an application of algorithm $\text{Refinement}(M_f, M, s_f)$ is defined as follows:

1. $S'_f$ is the set of states in $M_f$, possibly without the “failure” state $s_f$, but with the new states added by $\text{Refinement}()$. Then, $s'_0$ is the state in $S'_f$ such that $s_0 \in S'_f$ for $s_0 \in M$.

2. For $x \in \{\text{may}, \text{must}\}$, the transitions relations $\delta^x$ and the protocol functions $d^x_f$ are defined as in Def. 9. In particular,
   
   (a) for every $t, t' \in S'_f$ and joint action $\vec{a}, \vec{a}' \in \delta^\text{may}_f(t, \vec{a})$ iff for some $s \in t$ and $s' \in t'$, $\delta(s, \vec{a}) = s'$;
   (b) for every $t, t' \in S'_f$ and joint action $\vec{a}, \vec{a}' \in \delta^\text{must}_f(t, \vec{a})$ iff for all $s \in t$ there is $s' \in t'$ such that $\delta(s, \vec{a}) = s'$;
   (c) for every $t \in S'_f$ and $i \in Ag$, $d^x_{r_f}(i, t) = \{a_i \in \text{Act}_i \mid \delta^x_{f}(t, (a_i, \vec{a})) \text{ is defined for some } \vec{a}_2\}$.

3. For $v \in \{\text{tt}, \text{ff}\}$, $p \in AP$, and $t \in S'_f$, $V'_f(t, p) = v$ iff $V(s, p) = v$ for all $s \in t$; otherwise, $V'_f(t, p) = uu$.

By Def. 10 the components of the refined CGS $M'_f$ coincide with those in abstraction $M_f$, except possibly as regards the “failure” state $s_f$ and new states introduced by $\text{Refinement}()$. On the new states, the transition relations and protocol functions are defined in analogy with $M_f$.

We now show a property of the refined CGS $M'_f$, which will be useful to prove the main preservation result Theorem 3. Intuitively, must strategies in $M'_f$ respect uniformity on the set of their outcomes.
Theorem 3. In $M_t^*$, for every joint strategy $F_{\text{must}}^*$, for all $p, \bar{p} \in \text{out}(t, F_{\text{must}}^*)$, all $p' \in p, \bar{p}' \in \bar{p}$, and all $i \in \Gamma$, $j \in \mathbb{N}$, $p'_{t,j} \sim p_{t,j}$ then $F_{\text{must}}^*(p_{t,j}) = F_{\text{must}}^*(\bar{p}_{t,j})$.

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 $\Gamma$, its abstract CGS $M_{\Gamma}^*$ with refinement $M_t^*$, and state $s_{t,j} \ni s$, for every $\Gamma$-formula $\phi$ in $ATL^*$,

\[ ((M_{\Gamma}^*, s_{t,j}) \models \phi) \Rightarrow ((M, s) \models \phi) \tag{5} \]

\[ ((M_{\Gamma}^*, s_{t,j}) \not\models \phi) \Rightarrow (M, s) \not\models \phi \tag{6} \]

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

\[ \varphi = \langle \{t_1, c\} \rangle F (b \land \neg pUd) \]

becomes true as atom $b$ becomes defined in $a_1^1$ and $a_1^2$. So, by Theorem 3 formula $\varphi$ is true in the original iCGS $M$ as well.

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 $\Gamma$-formula $\phi$ in $ATL^*$, we first build the abstract, three-valued CGS $M_{\Gamma}^*$ as per Def. 9. We can model check $\phi$ on $M_{\Gamma}^*$, as the corresponding decision problem is decidable by Theorem 1, and then transfer 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 $\phi$ or any of its subformulas is undefined at some state $s_f$ in $M_{\Gamma}^*$, we can apply the refinement algorithm so as to obtain a refined CGS $M_{\Gamma}^*$; any defined value for $\phi$ on $M_{\Gamma}^*$ transfers to $M$ by Theorem 3. The refinement step can be iterated as long as $\phi$ 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