Agent-based Abstractions for Verifying Alternating-time Temporal Logic with Imperfect Information

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

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

ABSTRACT

We introduce a 3-valued abstraction technique for alternating-time temporal logic (ATL) in contexts of imperfect information. We introduce under- and over-approximations of an agent's strategic abilities in terms of *must*- and *may*-strategies and provide a 3-valued semantics for ATL based on agents in interpreted systems (IS). We define a relation of simulation between the agents and prove that it preserves defined truth values of ATL formulas. Finally, we introduce a notion of abstraction on IS and show that it simulates the concrete interpreted system. Under this setting we present a procedure that enables the direct construction of a finite abstraction from an infinite-state system.

1. INTRODUCTION

In recent years logic-based formalisms for representing and reasoning about strategic abilities, both individual and coalitional, have been a thriving area of research in formal methods for multi-agent systems [8, 6, 20]. Several multi-modal logics have been introduced to provide a formal account of complex strategic reasoning and behaviours for single agents and groups, including alternating-time temporal logic (ATL), coalition logic, and strategy logic [1, 10, 31, 32].

A key issue concerning these logics for strategic reasoning regards the development of techniques for verifying multi-agent systems (MAS) with respect to properties expressed in these rich formal languages. This endeavour is problematic as the typical verification problems for several of these formalisms, including model checking, are computationally harder than those for the temporal logics (e.g., CTL, LTL) that they subsume [15, 35].

Related to formal verification, a crucial distinction operated in game-theoretic contexts is whether players have perfect or imperfect information about other players and the environment they are interacting in and with. Originally, most of the logics for strategies mentioned above have been introduced in contexts of perfect information [1, 10], partly because this setting exhibits better computational properties. However, for many applications of interest, including autonomous agents, distributed computing, and economic theory, perfect information is either unrealistic as a working hypothesis or unattainable [16]. Imperfect information is known to make the verification task computationally more costly. As an example, while verifying ATL under perfect information is polynomial, the corresponding problem for imperfect information is Δ_2^P-complete [22]. When perfect recall is assumed, the problem goes from PTIME-complete to undecidable [15]. Thus, for logics of strategies to be adopted as specification languages in contexts of imperfect information, efficient verification tools and techniques, supporting model checking for practical cases of interest, should be developed.

In this paper we aim at contributing towards this long-term objective. Specifically, we introduce a novel 3-valued abstraction technique for the verification of ATL specifications with respect to (possibly infinite) MAS with imperfect information. Taking inspiration from [5, 19, 3, 4], we introduce under- and over-approximations of an agent's strategic abilities in terms of *must*- and *may*-strategies, then provide a 3-valued semantics for ATL based on agents in interpreted systems (IS). Next, we define a relation of simulation between agents and prove that it preserves defined truth values of ATL formulas. Finally, we introduce a notion of abstraction on IS and show that it simulates the corresponding concrete interpreted system. We illustrate the formal machinery by means of a toy example based on the Train Gate Controller [21], and conclude by discussing applications to the verification of strategic behaviours of agents in multi-agent systems.

Related work. The literature on abstraction-based techniques applied to the model checking problem has grown steadily in the past two decades [11, 12]. Here we focus on the contributions most closely related to the present work.

Multi-valued interpretations of modal logics has since long appeared in the literature [17, 18]. This approach has also been applied to the verification of temporal and epistemic logics [26, 27, 28], including ATL* under imperfect information [23]. In this line formulas are interpreted on some designated algebraic structure (possibly infinite) and modal operators correspond to operations on the values in the structure.

Here we also adopt a multi-valued semantics, 3-valued specifically, but our approach is essentially different from the references above, as it is based on the definition of under- and over-approximations of transition systems [5, 19]. Indeed, the main inspiration for this paper comes from [3, 34], which put forward 3-valued abstraction techniques for CTL and the alternating μ-calculus (AμC), assuming perfect information nonetheless. Abstractions are shown to preserve

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

defined truth values of formulas in the relevant logic, then a notion of pre-order between abstractions, akin to simulation, is introduced. A crucial difference here is that we consider imperfect information. Indeed, in AμC with perfect information, the ATL operators $\langle\langle\Gamma\rangle\rangle U$, $\langle\langle\Gamma\rangle\rangle G$, and $\langle\langle\Gamma\rangle\rangle F$ are definable in terms of the 'next' modality $\langle\langle\Gamma\rangle\rangle X$ and the fixed-point operators μ and ν. Hence, the technique in [3] deals directly only with the next fragment of AμC, then applies standard procedures for calculating fixed-points. However, this method fails when imperfect information is taken into account, as ATL operators can no longer be expressed via fixed points [7, 14]. Thus, original techniques have to be developed. Moreover, the semantics here proposed is agent-based, and so is the simulation relation and abstraction technique we develop, again differently from [3].

Recently, other 3-valued abstraction techniques have appeared [29, 30, 4], which differ from our account as regards the 3-valued semantics for ATL. Notably, here we introduce original notions of *may*- and *must*-strategies and prove that, differently from [30], our 3-valued semantics conservatively extends the standard 2-valued semantics for ATL.

Scheme of the paper. The rest of the paper is structured as follows. In Section 2 we present ATL and provide it with a semantics in terms of agent-based interpreted systems, suitable for MAS representation. In Section 3 we introduce a 3-valued semantics for ATL, based on general agents and general IS. Then, we show that, differently from [30], the 3-valued semantics conservatively extends the 2-valued one. In Section 4 we introduce simulation relations between agents and between IS, and prove that these indeed preserve the interpretation of ATL formulas (Theorem 1). Then, in Section 5 we define agent-based abstractions and prove that they simulate the original MAS (Theorem 2). We illustrate the formal machinery with an infinite version of the Train Gate Controller scenario.

2. INTERPRETED SYSTEMS WITH IMPERFECT INFORMATION

In this section we introduce the formal machinery that will be used throughout the paper. Hereafter we assume a set $Ag = \{1, \ldots, m\}$ of indexes for agents and a set AP of atomic propositions. Given a set U, $\overline{U}$ denotes its complement (w.r.t some $V \supseteq U$). Also, we denote the i-th element of a tuple v as either v_i or $v.i$.

Definition 1 (Agent). *Given a set Ag of agent indexes, an* agent *is a tuple $i = \langle L, Act, P, t\rangle$ such that*

- *L is the (possibly infinite) set of* local states*;*
- *Act is the (finite) set of* individual actions*;*
- *$P : L \to 2^{Act}$ is the* protocol function*;*
- *$t : L \times ACT \to L$ is the* local transition function*, where $ACT = Act_1 \times \cdots \times Act_{|Ag|}$ is the set of* joint actions*, s.t. for $l \in L$, $a \in ACT$, $t(l, a)$ is defined iff $a_i \in P(l)$.*

Intuitively, an agent i is situated in some local state $l \in L$, which represents the information she has about the whole system, and she can perform some action $a \in Act$, according to protocol P. Performing a joint action brings about a change in the state of the agent, according to transition function t. Hereafter we often identify an agent index i with the corresponding agent, the context will disambiguate. Also, we assume w.l.o.g. that for every local state $l \in L$, $P(l) \neq \emptyset$ by considering a null action *skip*, enabled in every local state, such that $t(l, a) = l$ whenever $a_i = skip$. Hence, the protocol P is a function from L to $2^{Act} \setminus \emptyset$.

Given a set Ag of agents, a *global state* $s \in \mathcal{G}$ is defined as a tuple $\langle l_1, \ldots, l_{|Ag|}\rangle$ of local states, one for each agent in Ag. Notice that an agent's protocol and transition function depend only on its local state, which might contain strictly less information than the global state s. In this sense agents have *imperfect information* about the system. This is in marked contrast with [3], where agents are assumed to have perfect information.

To describe formally the execution of a multi-agent system, we introduce the notion of interpreted system.

Definition 2 (IS). *An* interpreted system *is a tuple $M = \langle Ag, I, T, \Pi\rangle$ such that*

- *every $i \in Ag$ is an agent;*
- *$I \subseteq \mathcal{G}$ is the set of* (global) initial states*;*
- *$T : \mathcal{G} \times ACT \to \mathcal{G}$ is the* global transition function *such that $s' = T(s, a)$ iff for all $i \in Ag$, $s'_i = t_i(s_i, a)$;*
- *$\Pi : \mathcal{G} \times AP \to \{\mathrm{tt}, \mathrm{ff}\}$ is the* labelling function*.*

An interpreted system describes the interactions of a group Ag of agents, starting from some initial state in I, according to the transition function T. Atomic propositions are assigned values true (tt) or false (ff). Notice that the global transition function T is defined on global state s for joint action a iff $a_i \in P_i(s_i)$ for every $i \in Ag$. Finally, we introduce the set $\mathcal{S} \subseteq \mathcal{G}$ of global states reachable from set I of initial states, through the transition function T.

To reason about the strategic abilities of agents in interpreted systems, we make use of alternating-time temporal logic (ATL).

Definition 3 (ATL). *Formulas φ in ATL are defined by the following BNF, for $q \in AP$ and $\Gamma \subseteq Ag$:*

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

As customary, a formula $\langle\langle\Gamma\rangle\rangle\Phi$ is read as 'the agents in coalition Γ have a (collective) strategy to achieve Φ'. The meaning of LTL operators 'next' X, 'until' U, and 'always' G is standard. For a set $\Gamma \subseteq Ag$ of agents, we define Γ-formulas as the formulas where Γ is the only coalition appearing in ATL modalities.

Since the behaviour of agents in interpreted systems depends only on their local state, to provide an interpretation of ATL formulas on IS it is appropriate to consider a notion of strategy that takes into account local states only.

Definition 4 (Uniform Strategy). *A (*uniform, memoryless*)* strategy *for agent $i \in Ag$ is a function $f_i : L_i \to Act_i$ such that for every local state $l \in L_i$, $f_i(l) \in P_i(l)$.*

Intuitively, Def. 4 prescribes that in all global states s, s', in which the local state of agent i is the same, i.e., $s_i = s'_i$, a uniform strategy f_i returns the same action $f_i(s_i) = f_i(s'_i)$, in line with the standard account on uniform strategies [24]. Notice that we focus on memoryless (positional) strategy, for which only the current local state is relevant to select the action. The results below can in principle be extended

to memoryfull (perfect recall) strategies, but the technical details are more cumbersome. We leave this extension for future work.

Given an IS M, a *path* p is any infinite sequence $s_1 s_2 \dots$ of global states, in which p^i denotes the i-th element s_i. Further, for a set $F_\Gamma = \{f_i \mid i \in \Gamma\}$ of strategies, a path p is F_Γ-*compatible* iff for every $j \geq 1$, $p^{j+1} = T(p^j, a)$ holds for some joint action $a \in ACT$ such that for every $i \in \Gamma$, $a_i = f_i(p^j.i)$. Let $out(s, F_\Gamma)$ be set of all F_Γ-compatible paths starting from s.

We can now assign a meaning to ATL formulas on IS.

Definition 5 (Satisfaction). *The 2-valued satisfaction relation $\models^2$ for an IS M, state $s \in S$, and ATL formula ϕ is defined as follows (clauses for propositional connectives are immediate and thus omitted):*

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

A formula φ is *true* in an IS M, or $M \models^2 \varphi$, iff for all initial states $s \in I$, $(M,s) \models^2 \varphi$.

Definition 6 (Model Checking). *Given an IS M and an ATL formula ϕ, the model checking problem amounts to determining whether $M \models^2 \phi$.*

Observe that the semantics provided in Def. 5 is in line with similar proposals for interpreting ATL in contexts of imperfect information [24, 6]. In particular, the strategies generating paths are uniform, i.e., they only take into account the local state of agents and provide the same action in indistinguishable states. Further, here we adopt the *objective* interpretation of ATL with imperfect information, as opposed to the *subjective* interpretation whereby ATL operators are evaluated at state s against paths $p \in out(s', F_\Gamma)$, for all states s' indistinguishable from s for the agents in Γ, i.e., $out_s(s, F_\Gamma) = \bigcup_{i \in \Gamma, s'_i = s_i} out(s', F_\Gamma)$ [33]. Hereafter we deal primarily with the objective interpretation and discuss briefly the necessary modification concerning the subjective semantics.

The features of the above semantics entail that well-known principles, which are valid under perfect information, fail under imperfect information. Notably, the fixed-point characterisations of ATL operators $\langle\langle\Gamma\rangle\rangle G$, $\langle\langle\Gamma\rangle\rangle F$, and $\langle\langle\Gamma\rangle\rangle U$ do not hold (see [20] for counterexamples), thus making the model checking procedures based on them unusable and the verification task strictly more complex. In particular, ATL with imperfect information is not subsumed by the alternating μ-calculus [7, 14], as it is the case for perfect information, and therefore the techniques developed in [3] do not apply to the present setting. An original methodology will be developed in the next section.

We conclude this formal presentation with a toy example, whose unique purpose it to illustrate the machinery above.

2.1 The Train Gate Controller Scenario

We discuss a variant of the Train Gate Controller (TGC) scenario [21]. In the version we consider two trains, T_0 and T_1, try to access a tunnel, whose entrance is managed by controller C. Controller C allows only one train in the tunnel at any time and keeps track of how many times each train has accessed the tunnel by using counters c_0 and c_1. In case the trains are both requesting access, controller C grants it to the train with smaller count. If $c_0 = c_1$, then the choice is non-deterministic. By doing so, controller C tries to keep a fair access to the tunnel.

The formal specification for each train can be given as detailed in Def. 1, as follows:

Definition 7 (Train). *For $i \in \{0,1\}$, each train $T_i = \langle L_i, Act_i, P_i, t_i\rangle$ is defined as*

- $L_i = \{away, wait, tunnel\}$
- $Act_i = \{approach, enter, leave, skip\}$
- $P_i(away) = \{approach, skip\}$
  $P_i(wait) = \{enter\}$
  $P_i(tunnel) = \{leave\}$
- $t_i(away, a) = wait$ *if* $a_i = approach$
  $t_i(away, a) = away$ *if* $a_i = skip$
  $t_i(wait, a) = tunnel$ *if* $a_i = enter$ *and* $a_C = enter_i$
  $t_i(tunnel, a) = away$ *if* $a_i = leave$ *and* $a_C = leave_i$

By Def. 7 each train tries to access the tunnel in an execution loop. In particular, the entering and exiting from the tunnel are synchronised with controller C.

To model controller C we make use of four variables: *status*, with values green g and red r, counters c_0 and c_1 with range $\mathbb{N}$, and variable q with values 0, 1, and 2. Hereafter we use primed variables to denote their values at the next state. Variables not explicitly mentioned remain unchanged.

Definition 8 (Controller). *Controller $C = \langle L_C, Act_C, P_C, t_C\rangle$ is defined as*

- L_C *is the set of tuples* $(status, c_0, c_1, q)$ *s.t.* $Ran(status) = \{g, r\}$, $Ran(c_0) = Ran(c_1) = \mathbb{N}$, *and* $Ran(q) = \{0, 1, 2\}$
- $Act_C = \{enter_0, leave_0, enter_1, leave_1\}$
- *for every* $l \in L_C$ *and* $i \in \{0,1\}$,
  $P_C(l) = \{enter_i\}$ *if* $status = g$, $q = 2$ *and* $c_i < c_{1-i}$
  $P_C(l) = \{enter_0, enter_1\}$ *if either* $status = g$ *and* $q \neq 2$, *or* $status = g$, $q = 2$ *and* $c_0 = c_1$
  $P_C(l) = \{leave_0, leave_1\}$ *if* $status = r$
- $t_C(l, a) = l'$ *holds whenever* $status = g$, $a_C = enter_i$, $a_i = enter$, $status' = r$, $c'_i = c_i + 1$, *and* $q' = q - 1$; *or* $status = r$, $a_C = leave_i$, $a_i = leave$, $status' = g$; *or* $status = g$, $a_i = approach$, *and* $q' = q + 1$.

By Def. 8, controller C manages the entrance to the tunnel with semaphore *status*, which is synchronised with the entering and exiting of trains. Also, counters c_0 and c_1 keep track of how many times each train has accessed the tunnel. Finally, variables q (for queue) registers the number of trains in state *wait*. Notice that, since counters c_0 and c_1 take values in $\mathbb{N}$, controller C has infinitely many local states.

We now introduce the IS corresponding to this TGC scenario. In the following we consider a set AP containing atoms $c_0 < c_1$, $c_0 > c_1$, $c_0 = c_1$, in_tunnel_0, and in_tunnel_1, to be interpreted below.

Definition 9 (IS). *The IS for the Train Gate Controller scenario is the tuple* $M_{TGC} = \langle Ag, I, T, \Pi \rangle$ *such that*

- $Ag = \{T_0, T_1, C\}$*;*
- I *contains only the tuple* $(away, away, (g, 0, 0, 0))$*, i.e., both trains are away at the beginning, the semaphore is green, both counters are equal to 0, and no train is waiting;*
- T *is the composition of the local transition functions* t_i*, for* $i \in Ag$*, as per Def. 2;*
- *the labelling function* Π *is such that a comparison* $c_0 \star c_1$ *is true at state* s *iff the corresponding relation holds between the counters. Also,* $\Pi(s, in\_tunnel_i) = \mathrm{tt}$ *iff* $s_i = tunnel$*.*

We remark that, since controller C has infinitely many local states, the IS M_{TGC} is an infinite-state system.

Then, we can check, for instance, that in state $s = (wait, wait, (g, 0, 0, 2))$, for enabled joint action $a = (enter, enter, enter_0)$, we have transition $T(s, a) = (tunnel, wait, (r, 1, 0, 1))$. Therefore, we can already check that in state s controller C can (has a strategy to) let train T_0 into the tunnel, thus making c_0 greater than c_1:

$$(M, s) \quad \models^2 \quad \langle\langle C \rangle\rangle X(c_0 > c_1 \wedge in\_tunnel_0)$$

Moreover, in ATL we can specify various behaviours and strategic abilities of controller C and trains T_0 and T_1. As an example, we might state that, whenever the counter for train T_0 is smaller, she has a strategy to eventually make the counters equal.

$$AG((c_0 < c_1) \rightarrow \langle\langle T_0 \rangle\rangle F(c_0 = c_1)) \tag{1}$$

where AG is shorthand for $\langle\langle \emptyset \rangle\rangle G$.

Further, we can ask whether train T_0 has a strategy to engage the tunnel "infinitely often":

$$\langle\langle T_0 \rangle\rangle G \langle\langle T_0 \rangle\rangle F \; in\_tunnel_0 \tag{2}$$

Finally, we can express that, whenever the counters are equal, train T_0 and controller C have a strategy so that at the next step train T_1 has a strategy to make the counters equal:

$$AG((c_0 = c_1) \rightarrow \langle\langle C, T_0 \rangle\rangle X \langle\langle T_1 \rangle\rangle X(c_0 = c_1)) \tag{3}$$

Intuitively, formula (2) is true by the strategy whereby train T_0 keeps on engaging the tunnel; while the same strategy is not sufficient to make formula (1) true. Indeed, (1) is false in general. Also, (3) is true whenever controller C grants T_0's request, and then T_1 engage the tunnel.

In this TGC scenario we checked formulas (1)-(3) manually. However, we aim at developing an automated verification procedure capable of dealing with infinite-state systems, such as the IS M_{TGC}, for which standard model checking techniques cannot be immediately applied.

3. 3-VALUED SEMANTICS FOR ATL WITH IMPERFECT INFORMATION

This section is devoted to introducing a generalisation of the notion of agent in Def. 1 in terms of over- and under-approximations of their strategic abilities. Then, we present a 3-valued semantics for ATL, and show that this conservatively extends the 2-valued version in Section 2.

Definition 10 (Generalised Agent). *A* (generalised) agent *is a tuple* $i = \langle L, Act, P^{may}, P^{must}, t^{may}, t^{must} \rangle$ *such that*

- *sets* L *of local states and* Act *of actions are given as in Def. 1;*
- P^{may} *and* P^{must} *are* protocol functions *from* L *to* 2^{Act}*;*
- t^{may} *and* t^{must} *are* local transition relations *defined on* $L \times ACT \times L$ *such that, for* $x \in \{may, must\}$*,* $l \in L$*, and* $a \in ACT$*, transition* $t^x(l, a, l')$ *is defined for some* $l' \in L$ *iff* $a_i \in P^x(l)$*.*

By Def. 10 generalised agents have the same components of standard agents, but differently from Def. 1, we now distinguish between *may* and *must* protocols and transitions. This distinction can be understood in terms of approximations of the agents' abilities. Intuitively, the *may* protocol and transitions represent an over-approximation of these abilities, while *must* components can be seen as an under-approximation. Here the distinction between *must* and *may* components and related terminology derive from the literature [5, 19]. The intutive meaning will become apparent in Section 5, where we define agent abstractions. Moreover, the standard agents in Def. 1 are the limit case in which under- and over-approximations coincide, or formally, $P^{may} = P^{must}$ and $t^{may} = t^{must}$ is a function.

Hereafter we simply refer to agents, the context will disambiguate between generalised and standard agents. Further, we introduce a generalisation of the interpreted systems in Def. 2, in which atoms can be assigned a third truth value uu for 'undefined'. In what follows, for $x = may$ (resp. *must*), $\overline{x} = must$ (resp. *may*).

Definition 11 (Generalised IS). *A* (generalised) interpreted system *is a tuple* $M = \langle Ag, I, T, \Pi \rangle$ *such that*

- Ag *and* I *are given as in Def. 2;*
- $T : \mathcal{G} \times ACT \rightarrow \mathcal{G}$ *is the* global transition function *such that* $s' = T(s, a)$ *iff for all* $i \in Ag$*, either* $s'_i = t_i^{may}(s_i, a)$ *or* $s'_i = t_i^{must}(s_i, a)$*;*
- $\Pi : \mathcal{G} \times AP \rightarrow \{\mathrm{tt}, \mathrm{ff}, \mathrm{uu}\}$ *is the* labelling function*.*

The undefined value uu can be interpreted in various ways, for instance, unknown, unspecified, or inconsistent, depending on the application in hand. We do not discuss this matter further, as it is not relevant for our technical contribution. We say that the truth value τ is *defined* whenever $\tau \neq \mathrm{uu}$. If every agent in Ag is standard and the truth value of every atom is defined, then we say that the IS is standard as well, and we are back to Def. 2.

Given an IS M and a coalition $\Gamma \subseteq Ag$ of agents, we define an *indexed global transition relation* $T_\Gamma \subseteq \mathcal{G} \times ACT \times \mathcal{G}$ so that $T_\Gamma(s, a, s')$ holds iff (i) for all $i \in \Gamma$, $t_i^{must}(s_i, a, s'_i)$; and (ii) for all $i \in \overline{\Gamma}$, $t_i^{may}(s_i, a, s'_i)$. Intuitively, the transition relation T_Γ exhibits a conservative stance on the strategic abilities of coalition Γ (by considering the under-approximation t^{must}), and an optimistic view of the abilities of adversarial $\overline{\Gamma}$. Finally, we introduce the set $\mathcal{S} \subseteq \mathcal{G}$ of global states reachable from set I of initial states, through the transition relation T.

Next, as it was the case for *must* and *may* protocols and transitions, we introduce also *must*- and *may*-strategies.

DEFINITION 12 (UNIFORM x-STRATEGY). *For $x \in \{may, must\}$, a (*uniform, memoryless*) x-*strategy *for agent $i \in Ag$ is a function $f_i^x : L_i \to Act_i$ such that for every local state $l \in L_i$, $f_i^x(l) \in P_i^x(l)$.*

Here we distinguish between *may* and *must* strategies to over- and under-approximate the strategic abilities of agents. Again, the distinction collapse in the case of standard agents. Moreover, these notions can be extended to the case of memoryfull (perfect recall) strategies; again, we leave this for future work.

For a set $F_\Gamma^{must} = \{f_i^{must} \mid i \in \Gamma\}$ of strategies, a path p is F_Γ^{must}*-compatible* iff for every $j \geq 1$, $T_\Gamma(p^j, a, p^{j+1})$ holds for some joint action $a \in ACT$ such that for every $i \in \Gamma$, $a.i = f_i^{must}(p^j.i)$. Symmetrically, a path p is F_Γ^{may}*-compatible* if for every $j \geq 1$, $T_\Gamma(p^j, a, p^{j+1})$ holds for some joint action $a \in ACT$ such that for every $i \in \Gamma$, $a.i = f_i^{may}(p^j.i)$. For $x \in \{may, must\}$, let $out(s, F_\Gamma^x)$ be the set of all F_Γ^x-compatible paths starting from s.

DEFINITION 13 (SATISFACTION). *The 3-valued satisfaction relation $\models^3$ for an IS M, state $s \in \mathcal{S}$, and ATL formula ϕ is defined as in Fig. 1. In particular, in all other cases the value of ϕ is undefined (*uu*).*

Observe that, in the clauses for ATL operators, *must*-strategies are used to make formulas true, while *may*-strategies are used to falsify them. Further, we can introduce a 3-valued, subjective interpretation of ATL by defining in Def. 13 the set $out_s(s, F_\Gamma^x)$ of outcomes as $\bigcup_{i \in \Gamma, s'_i = s_i} out(s', F_\Gamma^x)$, for $x \in \{may, must\}$. Finally, $(M \models^3 \varphi) = \mathrm{tt}$ (resp. ff) iff for all (resp. some) $s \in I$, $((M, s) \models^3 \varphi) = \mathrm{tt}$ (resp. ff). Otherwise, $(M \models^3 \varphi) = \mathrm{uu}$.

We conclude the presentation of the 3-valued semantics with the following result, which shows that for standard IS it coincides with the 2-valued version. This is in contrast with recent proposals in this area [30, 29].

PROPOSITION 1. *In every standard IS M, for every state $s \in \mathcal{S}$ and ATL formula ϕ, the truth value $((M, s) \models^3 \phi)$ is always defined and*

$$((M, s) \models^3 \phi) = \mathrm{tt} \quad \textit{iff} \quad (M, s) \models^2 \phi$$
$$((M, s) \models^3 \phi) = \mathrm{ff} \quad \textit{iff} \quad (M, s) \not\models^2 \phi$$

PROOF. The proof is by induction on ϕ. If ϕ is an atom p, then $((M, s) \models^3 p)$ is defined as $\Pi(s, p)$ is. Further, $((M, s) \models^3 p) = \mathrm{tt}$ iff $\Pi(s, p) = \mathrm{tt}$, iff $(M, s) \models^2 p$. The case for $((M, s) \models^3 p) = \mathrm{ff}$ is similar. The cases for propositional connectives are immediate by the induction hypothesis. As for $\phi = \langle\langle\Gamma\rangle\rangle X\psi$, since for every $i \in Ag$, $P^{may} = P^{must}$ and $t^{may} = t^{must}$ is a function, the distinction between *may*- and *must*-strategies collapse. Hence, $((M, s) \models^3 \langle\langle\Gamma\rangle\rangle X\psi) = \mathrm{tt}$ iff for some strategy F_Γ, for all $p \in out(s, F_\Gamma)$, $((M, p^1) \models^3 \psi) = \mathrm{tt}$. By induction hypothesis we obtain that for some F_Γ, for all $p \in out(s, F_\Gamma)$, $(M, p^1) \models^2 \psi$, that is, $(M, s) \models^2 \phi$. The case for $((M, s) \models^3 \langle\langle\Gamma\rangle\rangle X\psi) = \mathrm{ff}$ is similar. To prove that the truth value of ϕ is defined, suppose that $((M, s) \models^3 \langle\langle\Gamma\rangle\rangle X\psi) \neq \mathrm{tt}$. That is, for every strategy F_Γ, for some $p \in out(s, F_\Gamma)$, $((M, p^1) \models^3 \psi) \neq \mathrm{tt}$. By induction hypothesis, the truth value of ψ is defined at each p^1, therefore $((M, p^1) \models^3 \psi) = \mathrm{ff}$. As a consequence of the satisfaction clause for falsehood, $((M, s) \models^3 \phi) = \mathrm{ff}$. The case for $((M, s) \models^3 \langle\langle\Gamma\rangle\rangle X\psi) \neq \mathrm{ff}$ is symmetric. The proofs for operators G and U are similar and therefore omitted. □

By Proposition 1 the 3-valued semantics for ATL is a conservative extension of the 2-valued semantics. Also, with minor modifications we can show that this result holds for the subjective interpretation of ATL as well.

4. SIMULATIONS

In this section we introduce a notion of simulation based on (generalised) agents that induces a simulation on (generalised) interpreted systems. Then, we show that simulations on IS preserve defined truth values of formulas in ATL. We begin by presenting simulation relations on local states. Here the crucial observation is that, since we aim at comparing interpreted systems where agents are defined on possibly different sets of local states and actions, simulations have to account for both these components.

DEFINITION 14 (LOCAL SIMULATION). *A* local simulation *for agent i is a pair (Σ_i, H_i) of relations $\Sigma_i \subseteq L_i \times L'_i$ and $H_i \subseteq Act_i \times Act'_i$ such that $\Sigma_i(l_1, l'_1)$ and $H_i(a_i, a'_i)$ imply*

1. *if $a_i \in P_i^{must}(l_1)$ then $a'_i \in P_i'^{must}(l'_1)$;*
2. *if $a'_i \in P_i'^{may}(l'_1)$ then $a_i \in P_i^{may}(l_1)$.*

Moreover, provided relations H_i as above for every $i \in Ag$, we write $H(a, a')$ for $H_i(a_i, a'_i)$ for every $i \in Ag$. Then, $H(a, a')$ implies

3. *for all $l_2 \in L_i$, if $t_i^{must}(l_1, a, l_2)$ then for some $l'_2 \in L'_i$, $t_i'^{must}(l'_1, a', l'_2)$ and $\Sigma_i(l_2, l'_2)$;*
4. *for all $l'_2 \in L'_i$, if $t_i'^{may}(l'_1, a', l'_2)$ then for some $l_2 \in L_i$, $t_i^{may}(l_1, a, l_2)$ and $\Sigma_i(l_2, l'_2)$.*

We say that local state l' *H-simulates* l, or $l \preceq_H l'$, iff $\Sigma(l, l')$ holds for some local simulation (Σ, H). Notice that, according to Def. 14 and by using standard terminology in reactive systems [25], if $l \preceq_H l'$ then l' 'simulates' *must*-transitions from l, while l 'simulates' *may*-transitions from l'. Hereafter, we assume the various H_i fixed for all agents $i \in Ag$, and, with an abuse of terminology, talk simply of simulation.

Given an agent $i = \langle L, Act, P^{may}, P^{must}, t^{may}, t^{must}\rangle$, we consider its primed version $i' = \langle L', Act', P'^{may}, P'^{must}, t'^{may}, t'^{must}\rangle$ defined on possibly different local states, actions, protocols, and transitions. We now introduce simulation relations on agents. Often, when clear by the context, we omit the prime, particularly in indexes of ATL operators.

DEFINITION 15 (AGENT SIMULATION). *The primed agent i' must-*simulates *agent $i \in Ag$ w.r.t. H, or $i \preceq_H^{must} i'$, iff*

1. *for every $a \in Act_i$, $H_i(a, a')$ for some $a' \in Act'_i$;*
2. *for every $a' \in Act'_i$, $H_i(a, a')$ for some $a \in Act_i$;*
3. *for every $l \in L$, $l \preceq_H l'$ for some $l' \in L'$.*

*Further, agent i' may-*simulates *i w.r.t. H, or $i \preceq_H^{may} i'$, iff (1) and (2) above hold, and*

3'. *for every $l \in L$, $l' \preceq_H l$ for some $l' \in L'$.*

Intuitively, agent i' *must*-simulates agent i (w.r.t. H) if i' has 'more' *must*-transitions and 'less' *may*-transitions than i. Symmetrically for *may*-simulations. Clearly, both $\preceq_H^{must}$

$((M,s) \models^3 q) = \tau$	iff	$\Pi(s,q) = \tau$, for $\tau \in \{\mathrm{tt}, \mathrm{ff}\}$
$((M,s) \models^3 \neg\psi) = \mathrm{tt}$	iff	$((M,s) \models^3 \psi) = \mathrm{ff}$
$((M,s) \models^3 \neg\psi) = \mathrm{ff}$	iff	$((M,s) \models^3 \psi) = \mathrm{tt}$
$((M,s) \models^3 \psi \wedge \psi') = \mathrm{tt}$	iff	$((M,s) \models^3 \psi) = \mathrm{tt}$ and $((M,s) \models^3 \psi') = \mathrm{tt}$
$((M,s) \models^3 \psi \wedge \psi') = \mathrm{ff}$	iff	$((M,s) \models^3 \psi) = \mathrm{ff}$ or $((M,s) \models^3 \psi') = \mathrm{ff}$
$((M,s) \models^3 \langle\langle\Gamma\rangle\rangle X\psi) = \mathrm{tt}$	iff	for some F^{must}_Γ, for all $p \in out(s, F^{must}_\Gamma)$, $((M,p^2) \models^3 \psi) = \mathrm{tt}$
$((M,s) \models^3 \langle\langle\Gamma\rangle\rangle X\psi) = \mathrm{ff}$	iff	for every F^{may}_Γ, for some $p \in out(s, F^{may}_\Gamma)$, $((M,p^1) \models^3 \psi) = \mathrm{ff}$
$((M,s) \models^3 \langle\langle\Gamma\rangle\rangle G\psi) = \mathrm{tt}$	iff	for some F^{must}_Γ, for all $p \in out(s, F^{must}_\Gamma)$, for all $k \geq 1$, $((M,p^k) \models^3 \psi) = \mathrm{tt}$
$((M,s) \models^3 \langle\langle\Gamma\rangle\rangle G\psi) = \mathrm{ff}$	iff	for every F^{may}_Γ, for some $p \in out(s, F^{may}_\Gamma)$, for some $k \geq 1$, $((M,p^k) \models^3 \psi) = \mathrm{ff}$
$((M,s) \models^3 \langle\langle\Gamma\rangle\rangle \psi U \psi') = \mathrm{tt}$	iff	for some F^{must}_Γ, for all $p \in out(s, F^{must}_\Gamma)$, for some $k \geq 1$, $((M,p^k) \models^3 \psi') = \mathrm{tt}$, and for all j, $1 \leq j < k$ implies $((M,p^j) \models^3 \psi) = \mathrm{tt}$
$((M,s) \models^3 \langle\langle\Gamma\rangle\rangle \psi U \psi') = \mathrm{ff}$	iff	for every F^{may}_Γ, for some $p \in out(s, F^{may}_\Gamma)$, for all $k \geq 1$, $((M,p^k) \models^3 \psi') = \mathrm{ff}$, or for some j, $1 \leq j < k$ and $((M,p^j) \models^3 \psi) = \mathrm{ff}$

Figure 1: The 3-valued satisfaction relation $\models^3$ for an IS M, state $s \in S$, and ATL formula ϕ.

and $\preceq^{may}_H$ are partial orders, i.e., reflexive and transitive relations, whenever H is such. Further, since agents in Def. 1 are a particular case of generalised agents (for which *may*- and *must*- protocols and transitions coincide), Def. 15 defines a relation of simulation for (non-generalised) agents as well. Hereafter, given a set $\Gamma \subseteq Ag$ of agents, we use $Ag'_\Gamma = \{i' \mid i \preceq^{must}_H i', i \in \Gamma\} \cup \{i' \mid i \preceq^{may}_H i', i \in \overline{\Gamma}\}$ to refer to the set of *must*- and *may*-simulating agents i', exactly one for each $i \in Ag$. Finally, a global state s' defined on Ag'_Γ *simulates* s on Ag (w.r.t. H), or $s \preceq^\Gamma_H s'$, iff (i) for every $i \in \Gamma$, $s_i \preceq_H s'_i$; (ii) for every $i \in \overline{\Gamma}$, $s'_i \preceq_H s_i$.

Definition 16 (IS Simulation). *Given a set $\Gamma \subseteq Ag$ of agents, an IS $M' = \langle Ag'_\Gamma, I', T', \Pi'\rangle$ Γ-simulates IS $M = \langle Ag, I, T, \Pi\rangle$ (w.r.t. H), or $M \preceq^\Gamma_H M'$, iff*

1. *Ag'_Γ defined as above is the set of simulations for agents in Ag;*
2. *for every $s \in I$, $s \preceq^\Gamma_H s'$ for some $s' \in I'$;*
3. *for every $s \in \mathcal{S}$, $s' \in \mathcal{S}'$, if $s \preceq^\Gamma_H s'$ and $\Pi'(s',p) = t$, for $t \in \{\mathrm{tt}, \mathrm{ff}\}$, then $\Pi(s,p) = t$.*

By Def. 16, if M' Γ-simulates M then every initial state in M is Γ-simulated by some initial state in M', and defined truth values of atoms are preserved from M' to M. Clearly, IS simulations are also partial orders, provided that H is. We observe that simulations for interpreted systems are indexed to sets of agents. Indeed this is normally the case for alternating simulations, as studied for instance in [2, 3].

The main result of this section shows that IS simulations preserve defined truth values of ATL formulas. To prove this, we need the following auxiliary lemma.

Lemma 1. *If $s \preceq^\Gamma_H s'$ then*

1. *for every strategy $F^{\prime must}_{\Gamma'}$, there exists strategy F^{must}_Γ such that for all $p \in out(s, F^{must}_\Gamma)$, there exists $p' \in out(s', F^{\prime must}_{\Gamma'})$ such that $p^k \preceq^\Gamma_H p'^k$ for every $k \geq 1$;*
2. *for every strategy F^{may}_Γ, there exists strategy $F^{\prime may}_{\Gamma'}$ such that for all $p' \in out(s', F^{\prime may}_{\Gamma'})$, there exists $p \in out(s, F^{may}_\Gamma)$ such that $p^k \preceq^\Gamma_H p'^k$ for every $k \geq 1$.*

Proof. As regards (1), suppose that $s \preceq^\Gamma_H s'$ and let $F^{\prime must}_{\Gamma'}$ be a strategy for primed coalition Γ'. We inductively define F^{must}_Γ on the length n of paths, and prove that it satisfies the statement of the lemma. For $n = 1$, for $i' \in \Gamma'$, consider $f^{\prime must}_i(s'_i) \in P^{\prime must}_i(s'_i)$ and $a_i \in P^{must}_i(s_i)$ such that $H_i(a_i, f^{\prime must}_i(s'_i))$: the existence of such a_i is guaranteed by Def. 14.2 and 15.2. We set $f^{must}_i(s_i) = a_i$ for $i \in \Gamma$. Further, if $T_\Gamma(s, a, p^2)$ for a extending $F^{must}_\Gamma(s_\Gamma)$, then (i) for all $i \in \Gamma$, $t^{must}_i(s_i, a, p^2_i)$; and (ii) for all $i \in \overline{\Gamma}$, $t^{may}_i(s_i, a, p^2_i)$. Then, by definition of local simulation, if $t^{must}_i(s_i, a, p^2_i)$ for $i \in \Gamma$, then for some $v'_i \in L'_i$, $t^{\prime must}_i(s'_i, a', v'_i)$ for a' extending $F^{\prime must}_{\Gamma'}(s'_{\Gamma'})$ in particular. Moreover, $s \preceq^\Gamma_H s'$ implies $s'_i \preceq_H s_i$ for every $i \in \overline{\Gamma}$. Again, by definition of local simulation, if $t^{may}_i(s_i, a, p^2_i)$ for $i \in \overline{\Gamma}$, then for some $u'_i \in L'_i$, $t^{\prime may}_i(s'_i, a', u'_i)$. Finally, let $p'^2 = (v_{\Gamma'}, u_{\overline{\Gamma}'})$. In particular, by construction we have that $T'_{\Gamma'}(s', a', p'^2)$ and $p^2_i \preceq^\Gamma_H p'^2_i$ for every $i' \in Ag'_\Gamma$. The inductive case is dealt with similarly. The proof for (2) is symmetric. □

Intuitively, Lemma 1 says that in Γ-similar states, a *must*-simulation $i' \in \Gamma'$ simulates *may*-strategies of $i \in \Gamma$, while her *must*-strategies are simulated by i. This can appear counterintuitive, but notice that in order to simulate *may*-strategies an agent must be capable of simulating *must*-transitions. Symmetrically for *may*-simulations and *must*-strategies.

By Lemma 1 we can prove our main preservation result.

Theorem 1. *If $M \preceq^\Gamma_H M'$, $s \preceq^\Gamma_H s'$ and $\tau \in \{\mathrm{tt}, \mathrm{ff}\}$, then for every Γ-formula ϕ,*

$$((M', s') \models^3 \phi) = \tau \quad \textit{implies} \quad ((M,s) \models^3 \phi) = \tau$$

Proof. The proof is by induction on the structure of ϕ. The base case for atoms follows by Def. 16.3 of IS simulation. The cases for propositional connectives are immediate. As regards ATL operators, we consider the case for $\phi = \langle\langle\Gamma\rangle\rangle X\psi$ being true. If $((M', s') \models^3 \phi) = \mathrm{tt}$ then for some strategy $F^{\prime must}_{\Gamma'}$, for all $q' \in out(s', F^{\prime must}_{\Gamma'})$, $((M', q'^2) \models^3 \psi) = \mathrm{tt}$. By Lemma 1.1, there exists strategy F^{must}_Γ such that for all $p \in out(s, F^{must}_\Gamma)$, for some $p' \in out(s', F^{\prime must}_{\Gamma'})$, $p^k \preceq^\Gamma_H p'^k$ for every $k \geq 1$. In particular, since for all $q' \in out(s', F^{\prime must}_{\Gamma'})$, we have that $((M', q'^2 \models^3 \psi) = \mathrm{tt}$, by induction hypothesis we obtain that for all $q \in out(s, F^{must}_\Gamma)$, $((M, q^2) \models^3 \psi) = \mathrm{tt}$ as well. As a result, $((M, q^2) \models^3 \phi) = \mathrm{tt}$. The case for $\phi = \langle\langle\Gamma\rangle\rangle X\psi$ being false follows by Lemma 1.2.

The inductive steps for formulas $\langle\langle\Gamma\rangle\rangle G\psi$ and $\langle\langle\Gamma\rangle\rangle \psi U \psi'$ are proved similarly, also by means of Lemma 1. □

By Theorem 1 we immediately obtain the following corollary.

COROLLARY 1. *If $M \preceq_H^\Gamma M'$, then for every Γ-formula ϕ,*

$$(M' \models^3 \phi) = \mathrm{tt} \quad \textit{implies} \quad (M \models^3 \phi) = \mathrm{tt}$$

By Corollary 1 a positive answer to the model checking problem for simulating IS M' entails a positive answer for simulated IS M as well. On the other hand, if ϕ is false in M', nothing can be derived about M.

We conclude this section by briefly discussing the modifications necessary to account for the subjective interpretation. Specifically, we require that $s \preceq_H^\Gamma s'$ implies (i) for every $v \in \mathcal{S}$, if $s_i = v_i$ for some $i \in \Gamma$, then $s'_i = v'_i$ for some $v' \in \mathcal{S}'$ such that $v \preceq_H^\Gamma v'$; and (ii) for every $v' \in \mathcal{S}'$, if $s'_i = v'_i$ for some $i \in \overline{\Gamma}$, then $s_i = v_i$ for some $v \in \mathcal{S}$ such that $v \preceq_H^\Gamma v'$. We state without proof that this modification is sufficient to prove Lemma 1, and then Theorem 1, for the subjective interpretation. The proofs follow similar lines of reasoning.

In the next section we apply the results above to derive finite, 3-valued abstractions of (possibly infinite) interpreted systems that preserve the defined truth value of ATL formulas.

5. ABSTRACTION

In this section we define an agent-based notion of abstraction for interpreted systems, indexed to a set Γ of agents. Then, we prove that abstractions Γ-simulate the original, concrete IS. Hence, they can be used in the verification procedure to make specific model checking instances more amenable.

To begin with, for every agent $i \in Ag$, let $\approx_i$ be an equivalence relation on L_i, and $[l] = \{l' \in L_i \mid l' \approx_i l\}$ be the equivalence class of l according to $\approx_i$. We first introduce abstractions for agents.

DEFINITION 17 (ABSTRACT AGENT). *Given an agent $i = \langle L, Act, P^{may}, P^{must}, t^{may}, t^{must}\rangle$ in Γ, the* abstract agent *$i' = \langle L', Act', P'^{may}, P'^{must}, t'^{may}, t'^{must}\rangle$ is defined such that*

1. *$L' = \{[l] \mid l \in L\}$;*
2. *$Act' = Act$;*
3. *$t'^{may}(l'_1, a, l'_2)$ iff for every $l_1 \in l'_1$, $t^{may}(l_1, a, l_2)$ for some $l_2 \in l'_2$; and $t'^{must}(l'_1, a, l'_2)$ iff for some $l_1 \in l'_1$, $l_2 \in l'_2$, $t^{must}(l_1, a, l_2)$;*
4. *for every $l' \in L'$, $P'^{may}(l') = \{a' \in Act \mid t'^{may}(l', a', l'_2)$ for some $l'_2 \in L'\}$ and $P'^{must}(l') = \bigcup_{l \in l'} P^{must}(l)$.*

On the other hand, if $i \in \overline{\Gamma}$, then i' is defined such that (1) and (2) holds and

3'. *$t'^{may}(l'_1, a, l'_2)$ iff for some $l_1 \in l'_1$, $l_2 \in l'_2$, $t^{may}(l_1, a, l_2)$; and $t'^{must}(l'_1, a, l'_2)$ iff for every $l_1 \in l'_1$, $t^{must}(l_1, a, l_2)$ for some $l_2 \in l'_2$;*
4'. *for every $l' \in L'$, $P'^{may}(l') = \bigcup_{l \in l'} P^{may}(l)$ and $P'^{must}(l') = \{a'_i \in Act \mid t'^{must}(l', a', l'_2)$ for some $l'_2 \in L'\}$.*

By Def. 17 the local states of abstract agent i' are the equivalence classes of local states for i; while the set of actions is the same. For $i \in \overline{\Gamma}$, the *may* (resp. *must*) protocol in an abstract local state l' includes all actions that are enabled in some (resp. all) concrete states $l \in l'$; while a *may* (resp. *must*) transition holds between abstract local states l'_1 and l'_2 iff from some (resp. all) concrete states $l_1 \in l'_1$, there is a transition to some $l_2 \in l'_2$. The definition for agent $i \in \Gamma$ is symmetric. Observe that Def. 17 fulfills the conditions on generalised agents in Def. 10. Moreover, whenever $i \in \overline{\Gamma}$ is a standard agent with $P^{may} = P^{must}$ and $t^{may} = t^{must}$ a function, we obtain that for the abstract agent i', $P'^{must} \subseteq P'^{may}$ and $t'^{must} \subseteq t'^{may}$; symmetrically for $i \in \Gamma$. This remark motivates the terminology of under- and over-approximations for *must*- and *may*-components. In particular, the role played by under- and over-approximations is symmetric for abstractions in Γ' and in $\overline{\Gamma}'$: when evaluating a Γ-formula we adopt a pessimistic stance of the strategic abilities of $i \in \Gamma$ and an optimistic view of $j \in \overline{\Gamma}$. Finally, notice that the abstraction i' of a standard agent i is not standard in general.

Hereafter, we do not make any specific assumption on the equivalence relation $\approx_i$ for agent i. In many cases of interest methodologies can be put forward to identify suitable equivalences. In the example below we briefly consider a simple form of predicate abstraction [13], by which states are equivalent iff they satisfy the same chosen predicates.

Next we prove that abstraction defines a simulation relation.

LEMMA 2. *For $i \in \Gamma$, the abstract agent i' must-simulates i. For $i \in \overline{\Gamma}$, the abstract agent i' may-simulates i.*

PROOF. Since conditions (1) and (2) in Def. 15 are trivially satisfied by the identity relation, we show that (3) and (3') hold as well. As regards (3) and *must*-simulations, we prove that the mapping $(l, a) \mapsto ([l], a)$ is a state simulation such that, for every $l \in L_i$, $l \preceq l'$ for $l' = [l]$. Remember that H_i be the identity relation for every $i \in Ag$. Then, $a_i \in P_i^{must}(l)$ implies $a_i \in P_i'^{must}(l') = \bigcup_{l \in l'} P_i^{must}(l)$, and $a'_i \in P_i'^{may}(l')$ implies that $t_i'^{may}(l', a', l'_2)$ for some $l'_2 \in L'$, that is, for every $l \in l'$, $a'_i \in P_i^{may}(l)$. Next, suppose that $t_i^{must}(l_1, a, l_2)$ for some $l_2 \in L$. Clearly $t_i'^{must}([l_1], a, [l_2])$ and $(l_2, a) \mapsto ([l_2], a)$. On the other hand, if $t_i'^{may}(l'_1, a, l'_2)$ and $(l_1, a) \mapsto (l'_1, a)$, then for some $l_2 \in l'_2$, $t_i^{may}(l_1, a, l_2)$ and $(l_2, a) \mapsto (l'_2, a)$. Thus, the mapping $(l, a) \mapsto ([l], a)$ witnesses the agent *must*-simulation $i \mapsto i'$. The proof for *may*-simulating agents is similar, this time by showing that $(l', a) \mapsto (l, a)$, for $l \in l'$, is a state simulation. □

Notice that the simulation relation H on actions in Lemma 2 is the identity relation, as concrete and abstract agents are defined on the same set *Act* of actions. This remark will simplify considerably the notation. In particular, we omit referring to H in simulations.

We can now introduce the notion of abstract interpreted system.

DEFINITION 18 (ABSTRACT IS). *Given an IS $M = \langle Ag, I, T, \Pi\rangle$ and a set Γ of agents, the abstract IS $M^\Gamma = \langle Ag^\Gamma, I^\Gamma, T^\Gamma, \Pi^\Gamma\rangle$ is defined as*

- *Ag^Γ is the set of abstractions of agents in Ag;*
- *$I^\Gamma = \{\langle [l_1], \ldots, [l_{|Ag|}]\rangle \mid \langle l_1, \ldots, l_{|Ag|}\rangle \in I\}$;*
- *for $\tau \in \{\mathrm{tt}, \mathrm{ff}\}$ and $p \in AP$, $\Pi^\Gamma(\langle [l_1], \ldots, [l_{|Ag|}]\rangle, p) = \tau$ iff for all $l'_i \in [l_i]$, $\Pi(\langle l'_1, \ldots, l'_{|Ag|}\rangle, p) = \tau$; otherwise, $\Pi^\Gamma(\langle [l_1], \ldots, [l_{|Ag|}]\rangle, p) = \mathrm{uu}$.*

By Def. 18 an atom p receives value true (resp. false) in an abstract state s' iff it receives such value in all corresponding concrete states s. Since the equivalence relations $\approx_i$ are not assumed to respect the assignment Π, some atoms might become undefined in some abstract states. Thus, the abstraction of a standard IS is not necessarily standard itself. However, here the aim is to trade definiteness with a smaller state space in M^Γ in comparison to M.

We can now prove the main result of this section.

THEOREM 2. *The abstraction M^Γ Γ-simulates the interpreted system M.*

PROOF. By Lemma 2, every abstract agent $i' \in Ag^\Gamma$ simulates agent $i \in Ag$. Further, for every $s \in I$, $s \preceq^\Gamma s'$ for $s' = \langle [s_1], \ldots, [s_{|Ag|}] \rangle$. Finally, for $s \in M$, $s' \in M'$, if $s \preceq^\Gamma s'$ and $\Pi'(s', p) = \tau$, then for every $l_i \in s'_i$, $\Pi(\langle l_1, \ldots, l_{|Ag|} \rangle, p) = \tau$. In particular, $\Pi(\langle s_1, \ldots, s_{|Ag|} \rangle, p) = \tau$. □

As an immediate consequence of Theorem 1 and 2, we obtain the following result.

COROLLARY 2. *If M^Γ is the abstraction of IS M, $s \in s'$, and $\tau \in \{\mathrm{tt}, \mathrm{ff}\}$, then for every Γ-formula ϕ,*

$$((M', s') \models^3 \phi) = \tau \quad \textit{implies} \quad ((M, s) \models^3 \phi) = \tau$$

By Corollary 2, under specific conditions, we can transfer the (defined) verification result from abstraction M' to concrete IS M. This result is of particular interest in all cases where M is infinite, while M' is finite, as in the following toy example. We conclude by stating that, with minor modifications, the abstraction procedure above can be applied to the subjective semantics as well.

5.1 The Abstract Train Gate Controller

Here we show how we can define a 3-valued abstraction for the infinite IS M_{TGC} in Section 2.1 that has the interesting feature of being finite. Specifically, by borrowing ideas from predicate abstraction for IS [30], we say that local states l and l' for agent i are *equivalent* iff they both satisfy the same local predicates. Then, notice that the only infinite component of the TGC scenario is the controller C herself, while both trains are finite. Further, in specifications (1) and (2) only train T_0 appears in ATL modalities, and therefore we set $\Gamma = \{T_0\}$. Now notice that each agent *must*- and *may*-simulates herself and both T_0 and T_1 are finite. Hence, we only need to define an abstract controller C^A that *may*-simulates C. To do so, we observe that the relevant predicates, appearing in the protocols and specifications, concern the values of c_0 and c_1 being either equal ($c_0 = c_1$) or one greater than the other ($c_0 < c_1$ and $c_0 > c_1$). Then, we introduce abstraction C^A for controller C in Def. 8, according to Def. 17, as follows:

DEFINITION 19 (ABSTRACTION C^A). *The abstract controller $C^A = \langle L_C^A, Act_C^A, P_C^A, t_C^A \rangle$ is given as*

- *L_C^A is the set of tuples $(status, comp, q)$ such that status and q are defined as for controller C, while $Ran(comp) = \{c_0 < c_1, c_0 > c_1, c_0 = c_1\}$*
- *the actions are the same as for controller C: $Act_C^A = Act_C$;*
- *the may- and must-protocol coincide: for every $l \in L_C^A$ and $i \in \{0, 1\}$,*

  $P_C^{Amay}(l) = \{enter_i\}$ if $status = g$, $q = 2$ and $comp = (c_i < c_{1-i})$;
  $P_C^{Amay}(l) = \{enter_0, enter_1\}$ if either $status = g$ and $q \neq 2$, or $status = g$, $q = 2$ and $comp = (c_0 = c_1)$;
  $P_C^{Amay}(l) = \{leave_0, leave_1\}$ if $status = r$;
- *we noticed that for abstractions of standard agents we have $t'^{must} \subseteq t'^{may}$. Hence, for every must-transition hereafter there is a corresponding may-transition (omitted):*

  $t_C^{Amay}((g, c_0 < c_1, q), (enter, a_1, enter_0), (r, c_0 < c_1, q-1))$
  $t_C^{Amay}((g, c_0 < c_1, q), (enter, a_1, enter_0), (r, c_0 = c_1, q-1))$
  $t_C^{Amay}((g, c_0 > c_1, q), (a_0, enter, enter_1), (r, c_0 > c_1, q-1))$
  $t_C^{Amay}((g, c_0 > c_1, q), (a_0, enter, enter_1), (r, c_0 = c_1, q-1))$
  $t_C^{Amust}((g, c_0 = c_1, q), (enter, a_1, enter_0), (r, c_0 > c_1, q-1))$
  $t_C^{Amust}((g, c_0 = c_1, q), (a_0, enter, enter_1), (r, c_0 < c_1, q-1))$
  $t_C^{Amust}((r, comp, q), (leave, a_1, leave_0), (g, comp, q))$
  $t_C^{Amust}((r, comp, q), (a_0, leave, leave_1), (g, comp, q))$
  $t_C^{Amust}((g, comp, q), (approach, a_1, a_C), (g, comp, q+1))$
  $t_C^{Amust}((g, comp, q), (a_0, approach, a_C), (g, comp, q+1))$

The key remark here is that, while the original IS M_{TGC} has infinitely many states, its abstraction M_{TGC}^A defined over trains T_0, T_1 and abstract controller C^A is finite with $|L_0 \times L_1 \times L_C^A| = 162$ global states. Hence, we can model check formulas (1)-(3) in Section 2.1, for instance, and then, in case we obtain a defined answer, transfer the result on the original IS M_{TGC} by means of Corollary 2.

6. CONCLUSIONS

In this paper we have introduced a 3-valued abstraction technique for infinite-state imperfect-information MAS specified via ATL. As discussed in the paper, and as it is to be expected in 3-valued abstraction, the technique cannot always resolve the value of a specification. To address this issue, in future work we plan to investigate refinement techniques [11] in contexts of imperfect information. Further, we plan to adapt the technique to support more powerful logics for strategic reasoning under incomplete information (e.g. Strategy Logic [9]. Finally, we envisage to implement the developed procedure in a model checking tool for the verification of strategic behaviours of agents.

Acknowledgements. This research was funded by EPSRC under grant EP/I00520X. F. Belardinelli acknowledges the support of the ANR JCJC Project SVeDaS (ANR-16-CE40-0021).

REFERENCES

[1] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.

[2] R. Alur, T. A. Henzinger, O. Kupferman, and M. Y. Vardi. Alternating refinement relations. In *Proceedings of the Ninth International Conference on Concurrency Theory (CONCUR'98)*, p. 163–178. Springer, 1998.

[3] T. Ball and O. Kupferman. An abstraction-refinement framework for multi-agent systems. In *Proceedings of the 21st Annual IEEE Symposium on Logic in Computer Science (LICS06)*, p. 379–388. IEEE, 2006.

[4] F. Belardinelli, A. Lomuscio, and J. Michaliszyn. Agent-based refinement for predicate abstraction of

multi-agent systems. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI16)*, p. 286–294. IOS Press, 2016.

[5] G. Bruns and P. Godefroid. Model checking partial state spaces. In *Proceedings of the 11th International Conference on Computer Aided Verification (CAV99)*, p. 274–287. Springer, 1999.

[6] N. Bulling, J. Dix, and W. Jamroga. Model checking logics of strategic ability. In *Specification and Verification of Multi-agent Systems*, p. 125–159. Springer, 2010.

[7] N. Bulling and W. Jamroga. Alternating epistemic mu-calculus. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI'11)*, p. 109–114. AAAI Press, 2011.

[8] N. Bulling and W. Jamroga. Comparing variants of strategic ability. *Autonomous Agents and Multi-Agent Systems*, 28(3):474–518, 2014.

[9] P. Cermák, A. Lomuscio, F. Mogavero, and A. Murano. MCMAS-SLK: A model checker for the verification of strategy logic specifications. In *Proceedings of the 26th International Conference on Computer Aided Verification (CAV14)*, p. 525–532. Springer, 2014.

[10] K. Chatterjee, T. Henzinger, and N. Piterman. Strategy logic. In *Proceedings of the 18th International Conference on Concurrency Theory (CONCUR07)*, p. 59–73, 2007.

[11] E. M. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith. Counterexample-guided abstraction refinement. In *Proceedings of the 12th International Conference on Computer Aided Verification (CAV00)*, p. 154–169. Springer, 2000.

[12] E. M. Clarke, O. Grumberg, and D. Long. Model checking and abstractions. *ACM Transactions on Programming Languages and Systems*, 16(5):1512–1542, 1994.

[13] S. Das, D. Dill, and S. Park. Experience with predicate abstraction. In *Proceedings of the 11th International Conference on Computer Aided Verification (CAV99)*, p. 160–171. Springer, 1999.

[14] C. Dima, B. Maubert, and S. Pinchinat. *Relating Paths in Transition Systems: The Fall of the Modal Mu-Calculus*, p. 179–191. Springer, 2015.

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

[16] C. Dwork and Y. Moses. Knowledge and common knowledge in a byzantine environment. *Information and Computation*, 88(2):156–186, 1990.

[17] M. Fitting. Many-valued modal logics. *Fundamenta Informaticae*, 15(3-4):335–350, 1991.

[18] M. Fitting. Many-valued modal logics II. *Fundamenta Informaticae*, 17:55–73, 1992.

[19] P. Godefroid and R. Jagadeesan. On the expressiveness of 3-valued models. In *Proceedings of the 4th International Conference on Verification, Model Checkig, and Abstract Interpretation (VMCAI03)*, p. 206–222. Springer, 2003.

[20] V. Goranko and W. Jamroga. Comparing semantics for logics of multi-agent systems. *Synthese*, 139(2):241–280, 2004.

[21] W. Hoek and M. Wooldridge. Cooperation, knowledge, and time. *Studia Logica*, 75(1):125–157, 2003.

[22] W. Jamroga and J. Dix. Model checking abilities under incomplete information is indeed Δ_P^2-complete. In *Proceedings of the 4th European Workshop on Multi-Agent Systems EUMAS'06*, p. 14–15, 2006.

[23] W. Jamroga, B. Konikowska, and W. Penczek. Multi-valued verification of strategic ability. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems (AAMAS16)*, p. 1180–1189. ACM, 2016.

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

[25] C. Baier and J. P. Katoen. *Principles of Model Checking*. The MIT Press, 2008.

[26] B. Konikowska and W. Penczek. Reducing model checking from multi-valued CTL* to CTL*. In *Proceedings of the 13th International Conference on Concurrency Theory (CONCUR02)*, p. 226–239. Springer, 2002.

[27] B. Konikowska and W. Penczek. Model checking multi-valued modal μ-calculus revisited. In *Proceedings of the International Workshop on Concurrency, Specification and Programming (CS&P04)*, p. 307–318. Humboldt University, 2004.

[28] B. Konikowska and W. Penczek. Model checking for multivalued logic of knowledge and time. In *Proceedings of the 5th international joint conference on Autonomous Agents and Multiagent Systems (AAMAS06)*, p. 169–176. IFAAMAS, 2006.

[29] A. Lomuscio and J. Michaliszyn. Verifying multi-agent systems by model checking three-valued abstractions. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS15)*, p. 189–198, 2015.

[30] A. Lomuscio and J. Michaliszyn. Verification of multi-agent systems via predicate abstraction against ATLK specifications. In *Proc. of the 15th Int. Conference on Autonomous Agents and Multiagent Systems (AAMAS16)*, 2016.

[31] F. Mogavero, A. Murano, G. Perelli, and M. Y. Vardi. Reasoning about strategies. *ACM Transactions in Computational Logic*, 15(4):34:1–34:47, 2014.

[32] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.

[33] P. Y. Schobbens. Alternating-time logic with imperfect recall. In *Proceedings of the International workshop on Logic and Communication in Multi-Agent Systems (LCMAS03)*, p. 1–12, 2004.

[34] S. Shoham and O. Grumberg. Monotonic abstraction-refinement for CTL. In *Proceedings of the 10th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS04)*, p. 546–560. Springer, 2004.

[35] G. van Drimmelen. Satisfiability in alternating-time temporal logic. In *Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science (LICS03)*, p. 208–213, IEEE Computer Society, 2003.