Decidable Verification of Multi-agent Systems with Bounded Private Actions

Extended Abstract

Francesco Belardinelli
Laboratoire IBISC, UEVE
Evy, France
belardinelli@ibisc.fr

Aniello Murano
Università degli Studi di Napoli
Naples, Italy
murano@na.infn.it

Alessio Lomuscio
Imperial College London
London, UK
a.lomuscio@imperial.ac.uk

Sasha Rubin
Università degli Studi di Napoli
Naples, Italy
rubin@unina.it

ACM Reference Format:

Introduction. A key feature of the frontier between decidability and undecidability in the verification of multi-agent systems (MAS) with respect to strategy-based specifications is the assumptions made on the agents’ information. For instance, the complexity of model checking MAS against ATL is markedly different for agents with complete information compared with agents with incomplete information, i.e., PTIME-complete [1] and undecidable [7], respectively, assuming agents with perfect recall. It follows that the verification problem for agents with incomplete information and perfect recall remains undecidable in any formalism stronger than ATL, such as Strategy Logic and most of its variants [4–6, 12]. It therefore remains of importance to identify expressive fragments whose model checking problem is decidable.

A way of achieving this consists in identifying classes of MAS, still endowed with perfect recall and incomplete information, for which model checking is decidable. For example, such a result is proved in [2, 3] for MAS where all communication is via public actions. In this paper we further pursue this line and show that the work in [2] can be generalised much further. Specifically, we show that: i) model checking remains decidable even if non-public actions are permitted a bounded number of times along any execution; ii) the temporal language underlying epistemic SL can be considerably extended from LTL to LDL [13] at no extra computational cost. By doing so, we show decidability for a much larger class of MAS against a considerably expressive specification language.

Interpreted Systems (IS) with Explicit Public Actions. Interpreted systems [8] are a formal setting for multi-agent systems (MAS), where each agent $a$ is defined by its set of local states $L_a$, set of actions $act_a$, local protocol function $P_a : L_a \to \mathcal{P}(act_a) \cup \{\emptyset\}$ (specifying available actions), and a local transition function $\tau_a : L_a \times Jact \to L_a$ where $Jact = \prod_a act_a$ is the set of joint actions.

This induces a transition system with state set $S = \prod_a L_a$, initial states in $S_0 \subseteq S$, and transition function $\tau : S \times Jact \to S$. An interpreted system $S$ is such a transition system with a valuation function $\pi : A\!P \to \mathcal{P}(S)$, where $A\!P$ is a set of atomic predicates.

A run (resp. history) is an infinite (resp. finite) sequence $r$ of global states such that $r(0) \in S_0$ and for every $n < |r|$ there exists a joint action $J \in Jact$ such that i) all agent actions are allowed by the respective individual protocols, and ii) $r(n+1) = r(n) J$. For $a \in Ag \text{ and } n < |r|$, let $r(a)(n)$ be the local state of agent $a$ in the $n$th global state of $r$. The set of all histories is denoted by Hist.

We now define a variant of IS in which some actions are public.

**Definition 1.** An interpreted system with explicit public actions is an IS such that, for every $a \in Ag$, there exists a set $pb\!_a \subseteq act_a$ of public actions, and a set $L_{pr\!_a}$ of private components such that:

i) $L_a = L_{pr\!_a} \times Jact$, where $Jact = \prod_a (pb\!_a \cup \{\emptyset\})$.

ii) The local transition function $\tau_a$ satisfies that $\tau_a((p, v), J) = (p', J')$ implies that for all $a \in Ag$, if $J_a \in pb\!_a$ then $J'_a = J_a$, else $J'_a = \varepsilon$.

iii) The initial global states are of the form $(l_a, (\varepsilon, \cdots, \varepsilon))_{a \in Ag}$.

That is, a local state is a pair $(p, v)$ where $p \in L_{pr\!_a}$ is the private local state proper, and $v \in Jact$ represents the last joint action performed in the system, with public actions only being visible and non-public actions represented by $\varepsilon$.

Any system conforming to Def. 1 is an IS. Also, any IS is isomorphic to some system conforming to Def. 1 (set $pb\!_a = \emptyset$ for all $a$). Thus, for convenience we will call systems conforming to Def. 1 simply IS. Also if $act_a = pb\!_a$ for all $a \in Ag$, we obtain systems with public actions only. These are closely related to the recording contexts in [8], game structures with public actions only [3], and MAS with broadcasting environment [10].

**Definition 2 (BMNPA).** A joint action $J \in Jact$ is called public if $J(a) \in pb\!_a$ for all $a \in Ag$. Let $\Delta$ denote the set of all public joint actions.

An interpreted system has only public actions after time $b$ if for every history $h \in Hist$, joint action $J \in Jact$, and $n \geq b$, if $r(h(n), J) = h(n+1)$ then $J$ is a public joint action. An interpreted system that has only public actions after time $b$, for some bound $b$, is said to have boundedly-many non-public actions; the class of such systems is denoted by BMNPA.
If an action is not in $pb\_act_a$, it may still be observed by all agents, e.g., by being recorded in their private components. Thus, $pb\_act_a$ should not be considered as the set of all public actions of agent $a$, but only those explicitly identified as such.

**Epistemic Dynamic Strategy Logic** We now define EDSL, an extension of Strategy Logic [12] in two directions: its temporal dimension, and agents. An assignment that maps a function $\chi$ states a strategy $\phi$ in the epistemic dimension.

**Definition 3 (EDSL).** The EDSL formulas $\phi$ and EDSL expressions $\rho$ over $AP$, $Ag$, and $Var$ are built according to the following grammar:

$$\phi ::= \rho \mid \neg \phi \mid \phi \land \psi \mid (\phi \psi) \mid (\chi \phi) \mid \langle x \rangle \phi \mid (x, a) \phi \mid C_A \phi \mid D_A \phi$$

$$\rho ::= \chi$$

where $\rho \in AP$, $x \in Var$, $a \in Ag$, $\phi \in Ag$, and $\chi$ is a propositional formula (i.e., a Boolean combination over $AP$).

**Semantics.** We interpret the logic EDSL on interpreted systems. To interpret the epistemic operators we introduce an indistinguishability relation $\sim_a$ on $S$, for every agent $a \in Ag$, such that $s \sim_a s'$ if $s(x) = s'(x)$ [8]. We extend $\sim_a$ to histories as follows: for $h, h' \in Hist$ define $h \equiv_a h'$ if $|h| = |h'|$ and $h(i) \equiv_a h'(i)$ for all $i \leq |h|$. Let $\equiv_A$ be $(\equiv \cup A \sim)$, where $\equiv$ denotes the reflexive and transitive closure (w.r.t. relation composition), and its extension to histories $\equiv^C_A$. Also, let $\sim_A$ be $\equiv^C_A \sim$, and let $\equiv^D_A$ be its extension to histories.

A strategy is a function of the form $\sigma : Hist \rightarrow Act$, and let $Str$ denote the set of all strategies. Strategy $\sigma$ is coherent for $a$ if action $\sigma(h)$ is available to $a$ in local state last(h)(a); it is uniform for $a$ if $h \equiv_a h'$ implies $\sigma_a(h) = \sigma_a(h')$, that is, in indistinguishable states $a$ is bound to play the same action [9]. An assignment is a function $\chi : Var \cup AG \rightarrow Str$ such that for every $a \in Ag$ the strategy $\chi(a)$ is coherent and uniform for $a$. For $x \in Var$ and $\sigma \in Str$, the variant $x^A_\sigma$ is the assignment that maps $x$ to $\sigma$ and coincides with $\chi$ on all other variables and agents. Similarly, if $a \in Ag$ and $\sigma$ is coherent and uniform for $a$, then the variant $x^a_\sigma$ is the assignment that maps $a$ to $\sigma \in Str$ and coincides with $\chi$ on all other variables and agents. An assignment $\chi$ is $\varphi$-compatible if, for every $x \in Var$, the strategy $\chi(x)$ is coherent and uniform for every agent in $shr(x, \varphi) = \{a \in Ag \mid (x, a)\varphi \}$ is a subformula of $\varphi$. Here $shr(x, \varphi)$ represents the set of agents using strategy $x$ in evaluating formula $\varphi$. We write out $(h, \chi)$ for the set of histories $h'$ generated by $\chi$, i.e., $h' \in out(h, \chi)$ if $h'$ is a prefix of $h'$ and for every $i \leq |h'|$, $h_{\leq i}' = (h_{\leq i}', \Pi_{a \in AG} \chi(a)(h_{\leq i}'_a))$.

**Definition 4 (Satisfaction).** Define the satisfaction relation $(S, h, \chi) \models \varphi$, where $h \in Hist$, $\varphi$ is an EDSL-formula, and $\chi$ is a $\varphi$-compatible assignment as follows:

$$\begin{align*}
(S, h, \chi) \models \pi & \quad \text{iff} \quad \text{last}(h) \in \pi(p), \text{for } p \in AP \\
(S, h, \chi) \models \neg \phi_1 & \quad \text{iff} \quad \text{it is not the case that } (S, h, \chi) \models \phi_1 \\
(S, h, \chi) \models \phi_1 \land \phi_2 & \quad \text{iff} \quad (S, h, \chi) \models \phi_1 \text{ for } i \in [1, 2] \\
(S, h, \chi) \models \langle x \rangle \phi_1 & \quad \text{iff} \quad \text{there exists a } \varphi_1\text{-compatible variant } \chi_2^A \text{ such that } (S, h, \chi_2^A) \models \phi_1
\end{align*}$$

It can be shown by induction that satisfaction is well-defined. In expressions $(S, h, \chi') \models \varphi'$, on the right-hand side assignment $\chi'$ is always $\varphi'$-compatible.
REFERENCES