Verifying Strategic Abilities in Multi-agent Systems with Private-Data Sharing

Extended Abstract

Francesco Belardinelli
Imperial College London and Université d’Evry
francesco.belardinelli@imperial.ac.uk

Catalin Dima
Université Paris-Est Créteil
dima@u-pec.fr

Ioana Boureanu
University of Surrey
i.boureanu@surrey.ac.uk

Vadim Malvone
Université d’Evry
vadim.malvone@univ-evry.fr

1 INTRODUCTION

Most formalisms for multi-agent systems (MAS) are not adept at explicitly expressing of data sharing between agents. Yet, disclosure and hiding of data amongst agents impacts on their strategic abilities, and so has a strong bearing on non-classical logics that formally capture agents’ coalitions, e.g., Alternating-time Temporal Logic (ATL) [1].

To this end, we devise concurrent game structures with propositional control for atom-visibility (vCGS). In vCGS, agents are and b have an explicit endowment to see some of each others’ variables, without other agents partaking in this. Second, we ascertain that the model checking problem for ATL with imperfect information and perfect recall on vCGS is undecidable. Third, we put forward a methodology to model check a formula \( \varphi \) in ATL* on a vCGS M, by verifying a suitable translation of \( \varphi \) in a submodel of M.

2 AGENTS WITH VISIBILITY-CONTROL

We introduce a class of systems where each agent can change the truth value of atoms she controls, and can make them (in)visible to other agents. On the underlying concurrent game structure, we interpret ATL*. We consider finite sets \( A\gamma \) and \( AP \) of agents and atoms.

Definition 2.1 (Visibility-Control Agents: Syntax). Given atom \( v \in AP \) and agent \( a \in A\gamma \), \( \text{vis}(v,a) \) denotes a visibility atom expressing intuitively that the value of \( v \) is visible to \( a \). By VA, we denote the set of all visibility atoms \( \text{vis}(v,a) \), for \( v \in AP \) and \( a \in A\gamma \). By \( \text{VA} = \{ \text{vis}(v,a) \mid v \in AP \} \), we denote the set of visibility atoms for agent \( a \).

Given set \( AP \) of atoms, an \( \text{agent} \) is a tuple \( a = \langle AP, V_a, G_C(a) \rangle \) s.t.

- \( V_a \subseteq AP \) is the set of atoms controlled by agent \( a \);
- \( G_C(a) \) is a finite set of guarded commands, which are of the form:

\[
\gamma := \varphi \quad \text{or} \quad \gamma := \gamma_1, \ldots, \gamma_k ;= \gamma_k
\]

where each \( \gamma_i \in V_a \) is an atom controlled by \( a \) that occurs at most once, \( \varphi \) is a boolean formula over \( AP \cup VA \), all \( a_1, \ldots, a_m \) are agents in \( A\gamma \) different from \( a \), and \( \text{tv} \) is a truth value in \( \{\text{tt}, \text{ff}\} \).

We denote with \( g(y) \) and \( \text{asg}(y) \) the guard and assignment, respectively. Moreover, guarded commands can be of two disjoint types: 1) \( \text{tt} \) - or update-type. In the former, the guard is always equal to \( \text{tt} \) (and thus omitted) and the assignment contains \( \text{vis}(v,a) := \text{tv} \) for every atom \( v \in V_a \) (i.e., an agent always has visibility of the atoms she controls).

Intuitively, Def. 2.1 says that (1) every agent \( a \) can change the value of the atoms in \( V_a \subseteq AP \) through assignments \( v := \text{tv} \); (2) agent \( a \) can switch the visibility for some other agent \( a_i \) over some of \( a \)'s atoms, by means of assignments \( \text{vis}(v,a) := \text{tv} \); (this is unlike [2, 18]); (3) since \( a_1, \ldots, a_m \) are required to be different from \( a \), agent \( a \) cannot remove visibility of her own atoms.

Hereafter, we assume that control is exclusive for any two distinct agents \( a \) and \( b \), \( V_a \cap V_b = \emptyset \), i.e., the sets of controlled atoms are disjoint. Since control is exclusive, we often talk about the owner of \( a \) in \( A\gamma \) for an atom \( v \in AP \).

Definition 2.2 (Visibility-Control Agents: Semantics). Given a set \( A\gamma \) of agents as in Def. 2.1, all defined on set \( AP \) of atoms, an iCGS with propositional control for atom-visibility (vCGS) is a tuple

\[
\mathcal{H} = \langle AP, \text{VA}, \text{Act}_A, \text{Act}_a, \text{G}_C, \gamma \rangle
\]

where:

- For every \( a \in A\gamma \), \( \text{Act}_a = \text{G}_C(a) \);
- \( S = \{ s \in \text{VA} \cup \text{VA} \mid \forall v \in AP \} \); whereas \( S_0 \subseteq S \) is the set of states \( s \) such that for every \( a \in A\gamma \), for some \( \gamma \) in \( \text{VA} \), if \( a \) is in \( S_0 \), if \( v := \text{tt} \) occurs in \( \text{asg}(y) \), then \( v \in V_a \); and if \( v := \text{ff} \) occurs in \( \text{asg}(y) \), then \( v \notin V_a \). That is, atoms are initialised as either true or false only via an \( \text{tt} \) command.

For every state \( s \in S \) and agent \( a \in A\gamma \), the protocol function \( P \) returns the set \( \text{P}(s,a) \) of update commands \( y \) such that \( \text{AP}(\gamma(s)) \subseteq \text{vis}(s,a) \) and \( s := g(y) \), where \( \text{AP}(\varphi) \) is the set of atoms occurring in formula \( \varphi \). That is, all atoms appearing in the guard are visible to the agent and the guard is indeed true.

The transition function \( \tau : S \times \text{Act} \rightarrow S \) is such that a transition \( \tau(s, (\gamma_1, \ldots, \gamma_k)) = s' \) holds iff (1) for every \( a \in A\gamma \), \( \gamma_a \in P(s,a) \); (2) for every \( v \in AP \) and \( \text{own}(v,a) \in \text{Act}(a) \) in \( s \) iff either \( \text{asg}(\gamma_{\text{own}(v)}) \) contains an assignment \( v := \text{tt} \) or \( v := \text{ff} \); whereas \( v \neq s' \) iff either \( \text{asg}(\gamma_{\text{own}(v)}) \) contains an assignment \( v := \text{ff} \) or \( v \neq s \); (3) for every \( v \in AP \) and \( \text{own}(v,a) \in \text{Act}(a) \) in \( s' \) iff either \( \text{asg}(\gamma_{\text{own}(v)}) \) contains an assignment \( \text{vis}(v,a) := \text{tv} \) or \( \text{vis}(v,a) \in s \); whereas \( \text{vis}(v,a) \neq s' \) iff either \( \text{asg}(\gamma_{\text{own}(v)}) \) contains an assignment \( \text{vis}(v,a) := \text{tv} \) or \( \text{vis}(v,a) \notin s \).

- Let the set \( R \subseteq S \) of reachable states be the transitive closure of \( S_0 \) under the transition function \( \tau \). The indistinguishability relation
is defined so that for every \( s, s' \in R, s \sim_a s' \) if \( \text{Vis}(s, a) = \text{Vis}(s', a) \) and for every \( v \in \text{Vis}(s, a) = \text{Vis}(s', a), v \in e \) iff \( v \in s' \); whereas for states in \( S \), each \( s \sim_a \) is the identity relation. We can easily check that if \( s \sim_a s' \) then \( P(s, a) = P(s', a) \).

- The labelling function \( \pi : S \to 2^{AP} \) is the identity, i.e., each state is named with the agents belonging to it.

**ATL Syntax.** State (\( \varphi \)) and path (\( \psi \)) formulas in ATL* are defined as follows, where \( q \in AP \) and \( A \subseteq Ag: \varphi ::= q | \neg \varphi \lor \varphi \land (\langle A \rangle \psi); \psi ::= \varphi | \neg \psi | \psi \land \varphi | X \psi | (\psi \rightarrow \varphi) \). Formulas in the ATL fragment of ATL* are obtained by restricting path formulas \( \psi \) as follows, where \( \varphi \) is a state formula: \( \psi ::= X\varphi | (\varphi \cup \psi) | (\varphi \land \psi) \).

**ATL Semantics.** Given a vCGS \( M \), a path \( p \) is a sequence \( s_1s_2... \) of states such that for every \( i > 1 \) there exists a joint action \( a \in ACT \) such that \( (s_i, a) \in i, s_{i+1} \). A finite path \( h \in S_0 \) starting in an initial state is called a history. Hereafter, we extend the indistinguishability relation \( \sim \) to histories in \( S_0 \) in a similar way and pointwise as follows, that is, \( h \sim h' \) if \( |h| = |h'| \) and for every \( i \leq |h|, h_i \sim h_i' h \). A uniform, memoryful strategy for agent \( a \in AG \) is a function \( f_a : S_0 \to A^{\Delta t} \) such that for all histories \( h, h' \in S_0 \), (i) \( f_a(h)(p) = f_a(h')(p) \) if \( h_i = h_i' \); (ii) if \( h_i \neq h_i' \) then \( f_a(h)(p) = f_a(h')(p) \). A joint strategy \( \Delta t \) for agent \( a \in A \) for coalition \( A \subseteq AG \) and history \( h \in S_0 \), let \( \text{out}(h, F_A) \) be the set of all infinite paths \( p \) starting from history \( h \) and compatible with \( F_A \). More formally, we set \( \text{out}(h, F_A) = \{ p | p \leq |h| = \# a \) and for all \( i \geq |h|, p_{i+1} = \tau(p_i) \), where for all \( a \in A, a_0 = \tau(p_{i+1}) \).

The satisfaction relation \( \models \) for a vCGS \( M \), path \( p \), index \( i \in H \), and ATL* formula \( \varphi \) is defined as follows (clauses for Boolean operators and temporal operators are immediate and thus omitted): (1) \( \langle M, p, i \rangle, |q \models q \in i; (2) \langle M, p, i \rangle, |(\langle A \rangle \psi) \models q \iff \exists m (\langle M, m, i \rangle, |q \models (\psi \rightarrow \varphi); (3) \langle M, p, i \rangle, |(\psi \land \varphi) \models q \iff (\langle M, p, i \rangle, |q \models \varphi \). A formula \( \varphi \) is satisfied by a vCGS \( M \), or \( M \models \varphi \), iff for all paths \( p \) starting in an initial state, \( (M, p, 1) \models \varphi \). Note that we adopt the objective interpretation of ATL* [15], whereby strategy operator \( (\langle A \rangle) \) is evaluated against all paths \( p \in \text{out}(h, F_A) \) starting from the present history. The checking problem for vCGS agents asks for checking whether a given vCGS satisfies a given ATL* formula \( \varphi \).

We state the main result of this section.

**Theorem 3.2.** The model checking problem for ATL* (resp. ATL) on vCGS is undecidable.

Theorem 3.2 is proved by showing that model checking ATL on standard iCGS [15], which is known to be undecidable [10], is PTIME-reducible to the same problem on vCGS. A proof can be found in [5].

### 3 FORMULA-BASED MODEL REDUCTION

Now, we put forward a methodology to model check a formula \( \varphi \) in ATL* on a vCGS \( M \) by verifying a suitable translation of \( \varphi \) in a submodel of \( M \). This reduction leads in general to a smaller state space and a less complex model checking instance. Under specific circumstances it might lead to decidability model checking.

**Definition 3.1.** Given a vCGS \( M \) and formula \( \varphi \), we define by mutual recursion the sets \( \Delta \subseteq AP \) of atoms and \( \Gamma \subseteq AG \) of agents:

\[
\begin{align*}
\Delta_0 &= \{ \varphi \} \\
\Delta_{n+1} &= \Delta_n \cup \{ AP(g(y)) | \text{some } v \in \Delta_n \text{ appears in } asg(y) \} \\
\Gamma_{n+1} &= \Gamma_n \cup \{ v \in \Delta_n | v \in \text{Act} \}
\end{align*}
\]

where we recall that \( AP(\varphi) \subseteq AP \) is the set of atoms appearing in formula \( \varphi \), and we take \( \text{Own}(\Delta_i) \) to be \( \{ \text{own}(v) \mid v \in \Delta_i \} \subseteq AG \). Then, let \( \Delta = \bigcup_{n \in \mathbb{N}} \Delta_n \subseteq AP \) and \( \Gamma = \bigcup_{n \in \mathbb{N}} \Gamma_n \subseteq AG \).

Intuitively, \( \Delta \) is the set of all atoms that are relevant to determine the atoms in formula \( \varphi \), the visibility of agents in \( \Gamma \), or of action guards that influence the truth of atoms in \( \Delta \) (including \( AP(\varphi) \)); whereas \( \Gamma \) is the set of owners of the atoms in \( \Delta \).

**Definition 3.2.** Given \( \Delta \) and \( \Gamma \) as in Def.3.1 and an agent \( a \in \Gamma \), we define a new agent \( a' = (\Delta, \text{Own}(\Delta), \text{Act}(\Delta)) \) such that:

\[
\begin{align*}
V_{a'} &= V_a \cap \Delta; \\
g_{a'}(y) &= g(y) \cdot \text{asg}(y) \cdot \text{Act}(\Delta) \\
\bigcup v \in AP | \text{vis}(v, a) \text{ appears in } g(y) \big) \subseteq \Delta; \\
\text{where } \text{asg}(y)(\Delta, \Gamma) \text{ is the component-wise restriction of } p \text{ to } \Delta \text{ and } \Gamma.
\end{align*}
\]

We now state the main result of this section. First of all, given a model \( M \), we write \( M_{\Delta, \Gamma} \) for the model generated by using agents as in Def.3.2 and restricted over \( \Delta \). Further, given a formula \( \varphi \), we write \( \varphi_{\Delta, \Gamma} \) for the formula generated by substituting every sub-formula \( (\langle A \rangle) \) of \( \varphi \) with \( (\langle \Delta \cap A \rangle) \). Also, given a path \( p, p_{\Delta, \Gamma} = (p_1)_{\Delta, \Gamma}, (p_2)_{\Delta, \Gamma}, \ldots \), \( (\langle \Delta \cap A \rangle) \) is the component-wise restriction of \( p \) to \( \Delta \) and \( \Gamma \).

**Theorem 3.3.** Given a vCGS \( M \) and a formula \( \varphi \), we have that \( (M, p, i) \models \varphi \iff (M_{\Delta, \Gamma}, p_{\Delta, \Gamma}, i) \models \varphi_{\Delta, \Gamma} \)

By Theorem 3.3 we obtain the following corollary.

**Corollary 3.4.** Given a vCGS \( M \) with reduction \( M_{\Delta, \Gamma} \), and a formula \( (\langle \Delta \rangle) \), \( \varphi_{\Delta, \Gamma} \), if \( \Delta \subseteq \Delta \) then it is decidable whether \( M \models \varphi \).

### 4 RELATED WORK AND CONCLUSIONS

Coalition logic based on propositional control [20] has been extended with transfer of control in [4, 11, 12, 19]. Yet, these mainly deal with coalition logic, which is the “next” \( (\langle A \rangle) \) fragment of ATL, while assuming perfect information. Meanwhile, we analyse the case of imperfect information and for the whole of ATL*. This is also unlike verification of just game-theoretic equilibria under imperfect information [13, 14], in reactive modules with guarded commands [2]. Similarly, other works miss the strategic-ability edge that we have, yet they focus on mere expressivity, e.g., at the epistemic level. This is the case of [16] where (propositional) visibility is also analysed but employing modal operators for visibility. Only limited type of strategic reasoning, stemming from no local “implementation” of actions, is also offered by another semantics similar to ours, i.e., dynamic epistemic logic and epistemic planning [21]. Finally, various restrictions on iCGS to retain decidability of model checking under imperfect information and perfect recall has been recently explored in [5–9, 17].

We put forward a formalism for the explicit expression of private-data sharing in multi-agent systems. On these “MAS with 1-to-1 private-channels”, we ascertain that the model checking problem for Alternating-time Temporal Logic under imperfect information and perfect recall is, as expected, undecidable. Yet, we put forward a methodology to model check a formula \( \varphi \) in ATL* on a vCGS \( M \) by verifying a suitable translation of \( \varphi \) in a submodel of \( M \). As future work, we aim to find general classes of vCGS for which the model checking of ATL becomes decidable, and show-case vCGS in modelling CIT problems.

Acknowledgements. F. Belardinelli acknowledges the support of ANR JCJC Project SVeDaS (ANR-16-CE40-0021).
REFERENCES