Verification of Multi-agent Systems with Imperfect Information and Public Actions

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

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

Aniello Murano
and Sasha Rubin
DIEI
Università degli Studi di Napoli
murano@na.infn.it
rubin@unina.it

ABSTRACT
We analyse the verification problem for synchronous, perfect recall multi-agent systems with imperfect information against a specification language that includes strategic and epistemic operators. While the verification problem is undecidable, we show that if the agents’ actions are public, then verification is 2EXPTIME-complete. To illustrate the formal framework we consider two epistemic and strategic puzzles with imperfect information and public actions: the muddy children puzzle and the classic game of battleships.

1. INTRODUCTION
Synchronous, perfect-recall multi-agent systems (MAS) are an important class of MAS that can be used to model a wide variety of scenarios including communication protocols, security protocols and games [8]. Reasoning about the knowledge and the strategic ability of agents in these systems remains of particular importance. Traditionally, epistemic logic [8] has been used to express the states of knowledge of the agents, whereas ATL has provided a basis for the agents’ strategic abilities [1]. ATL and epistemic logic have been combined in a number of ways to obtain specification languages capable of expressing both concepts (see below). A popular method for establishing properties of MAS is verification via model checking [4].

However, verifying synchronous, perfect recall MAS under incomplete information against specifications in ATL is undecidable [1, 6] (hence it remains undecidable when epistemic modalities are added); it is therefore of interest to identify cases in which reasoning about MAS is decidable. These restrictions typically take three forms: restricting the syntax of the logic (e.g., by removing strategic abilities and consider, instead, the extension of LTL with individual-knowledge operators, as in [32]), restricting the semantics (e.g., by requiring strategy quantifiers to vary over memoryless-strategies [31]), or by restricting the class of MAS under consideration. In this paper we follow the third option.

Contribution. We identify a class of imperfect-information concurrent game structures (iCGS) that we call public-action iCGS (PA-iCGS). In contrast to general iCGS [6], we prove that model-checking the full logic ATL^{K,E,D}_\ast on PA-iCGS is decidable, specifically 2EXPTIME-complete. Thus, the joint complexity of model-checking is the same as that of ATL^\ast with perfect information [1]. Moreover, we show that the class models MAS in which agents have imperfect information, synchronous perfect recall, and whose actions are public, i.e., all actions are visible to all agents. As we explain, the class PA-iCGS captures games of imperfect information in which the agents have uncertainty about the initial configuration but all moves are observable to all agents. This has applications to, among others, games (e.g., Bridge, Poker, Battleships, etc.), fair division protocols (e.g., classic cake cutting algorithms), selected broadcast protocols [33], blackboard systems in which a public database is read and written by agents [26], auctions and auction-based mechanisms [7].

The rest of the paper is organised as follows. In the remainder of this section we discuss related work. In Section 2 we define iCGS with public actions and the logic ATL^{K,E,D}_\ast that we will use as specification language and illustrate the formalism. In Section 3 we present the main result of the paper, i.e., we show the decidability of the verification problem, by means of an automata-theoretic approach, and analyse the resulting complexity. In Section 4 we compare our approach to that of Broadcast Environments [33]. We conclude in Section 5.

Related Work. In order to reason formally about multi-agent systems, temporal logics such as LTL, CTL, CTL^\ast have been extended with strategy quantifiers [1] and epistemic modalities [14]. The extended syntax has been combined with a number of different assumptions on the underlying MAS: perfect vs. imperfect information, perfect vs. imperfect recall, state-based vs. history-based semantics [1, 14, 10, 31, 15, 5, 11, 25].

Assuming imperfect information and perfect recall, as we do in this paper, often results in intractable model-checking. For instance, the model-checking problem for ATL in this setting is undecidable [6], as is the model-checking problem for the extension LTL^{K,C}_\ast of LTL with epistemic operators, including common knowledge [32]. Given this difficulty, finding decidable or tractable fragments remains of interest.

As expected, restricting the logic lowers the complexity. We list some notable examples: the model-checking problem for LTL with only individual knowledge is non-elementary complete [32], model-checking ATL with only “communicating coalitions” (i.e., coalitions use their distributed knowledge instead of their individual knowledge) is decidable and
non-elementary [5, 11]; and, model-checking ATL in which
all coalitions operate with a single indistinguishability rela-
tion reduces ATL to its singleton-coalition fragment [17].

Also, restricting the class of structures (iCGS) over which
these languages are interpreted lowers the complexity. Such
restrictions typically take one of two forms: i) on the obser-
vation or information sets of the agents; ii) on the the ar-
chitectures that govern communication. Notable examples
of i) may require that: all agents have the same observation
sets [21]; that the information sets form a hierarchy [27], or
that, over time, they infinitely often form a hierarchy [2].
A notable example of ii) are characterisations of the archi-
tectures for which distributed synthesis is decidable [9, 30],
thus generalising earlier results on linear architectures [27, 18].

More closely related to the present contribution are broad-
cast environments (which restrict the underlying iCGSs) and
that can capture epistemic puzzles and games of imperfect
information such as Bridge [33]. The most relevant result for
broadcast environments is that synthesis of linear-temporal
logic with individual-knowledge operators is decidable [33].
Not only can our language express this synthesis problem,
but it is strictly more expressive, as it can alternate strat-
egic quantifiers mentioning overlapping coalitions. A detailed
discussion of the significance of [33] is given in Section 4.

Actions that constitute public announcements have been
studied in depth (Dynamic Epistemic Logic, Public An-
nouncement Logic, epistemic protocols [34]). However, this
line of research differs semantically and syntactically from
our work. In particular, in these works modal operators are
model transformers, and coalitions are not explicitly named
in the language.

2. GAMES WITH PUBLIC ACTIONS AND
STRATEGIC-EPISTEMIC LOGIC

In this section we define the game model and the logic.
The model is the subclass of imperfect information concurrent
game structures (iCGSs) that only have public actions
(PA-iCGS). The logic is ATL
X ∪ Ω, an extension of Alternating
Time Temporal Logic (ATL
X ) which includes strategic operators
(⟨⟨A⟩⟩) for A ⊆ Ag) as well as epistemic operators for
individual-knowledge (K
a for a ∈ Ag), common-
knowledge (C
a for A ⊆ Ag), and distributed-knowledge (D
a for
A ⊆ Ag).

Notation. For an infinite or non-empty finite sequence u ∈ X
X ∪ X
X the empty sequence is denoted ϵ. The length of a
finite sequence u ∈ X
u is denoted |u|, its last (resp. first) element
is denoted last(u) (resp. first(u)). Note that last(ϵ) =
first(ϵ) = ϵ. For i < |u| we write u
u ∈ X, for the prefix u
u, u ∈ X.X for a vector v ∈ X
v we denote the i-th co-ordinate of v by v(i). In particular,
for F ∈ ∏(X
0, we may write F(i) ∈ X
F(i)(u) ∈ X.

2.1 iCGS with only Public Actions

We begin with the standard definition of imperfect infor-
mation concurrent game structures [3, 6].

Definition 1 (iCGS). An imperfect information con-
current game structure (iCGS) is a tuple

S = (Ag, AP, Α, δ, (∼a) a ∈ Ag, λ)

where:

- Ag is the finite non-empty set of agent names;
- AP is the finite non-empty set of atomic propositions;
- Α is the finite non-empty set of actions for a ∈ Ag;
- S is the finite non-empty set of states;
- S0 ⊆ S is the non-empty set of initial states;
- δ : S × Α → S is the transition function, where the
set Α of joint-actions is the set of all functions
J : Ag → ∪a Α such that J(a) ∈ Αa. The transition
function assigns to every state s and joint-action J, a
successor state δ(s, J);
- ∼a ⊆ 2S is the indistinguishability relation for agent a,
which is an equivalence relation; the equivalence class
[s]a of s ∈ S under ∼a is called the observation set of
agent a;
- λ : AP → 2S is the labeling function that assigns to
each atom p the set of states λ(p) in which p holds.

Perfect-information is treated as a special case:

Definition 2 (perfect-information). A concurrent
game structure (CGS) is an iCGS for which ∼a = (s, s) :
for all a ∈ Ag.

We now give a brief and to-the-point definition of what
it means for an iCGS to only have public actions, i.e.,
all actions are visible to all agents. This determines a subclass
of iCGSs that we call PA-iCGS.

Definition 3 (PA-iCGS). An iCGS only has public
actions if for every agent a ∈ Ag, states s, s′ ∈ S, and
joint actions J, J′ ∈ Α, if J ̸= J′ and s ∼a s′ then
δ(s, J) ̸= δ(s′, J′). We write PA-iCGS for the class of
iCGS that only have public actions.

This definition says that if an agent a cannot distinguish
between two states, but different joint actions are performed
in each of these states (because, for instance, some other
agent can distinguish them), then the agent can distinguish
between the resulting successor states.

One way to generate an iCGS only having public actions is
to ensure that i) the state records the last joint-action
played, thus S is of the form T × (Α ∪ {ε}), where
ε refers to the situation that no actions have yet been played,
and ii) the indistinguishability relations ∼a satisfy that if
(t, J) ∼a (t′, J′) then J = J′. Similar conditions have been
considered in the literature, e.g., recording contexts in [8].

In the remainder of this section we define what it means
for an agent to have synchronous perfect-recall [8].

Synchronous perfect-recall under imperfect informa-
tion. A path in S is a non-empty infinite or finite sequence
π = π1 · · · ∈ S ∪ S
T such that for all t there exists a joint-
action J(t) ∈ Α such that πt+1 ∈ δ(πt, J(t)). Paths that
start with initial states are called histories if they are finite
and computations if they are infinite. The set of computa-
tions in S is written comp(S), and the set of computations in
S that start with history h is written comp(S, h). We define
hist(S) and hist(S, h) similarly.

We use the following notation: if ∼ is a binary relation
on S we define the extension of ∼ to histories as the binary
relation ≡ on hist(S) define by h ≡ h′ if |h| = |h′| (i.e., syn-
chronicity) and hj ∼ h′j for all j ≤ |h| (i.e., perfect recall).
We give three particular instantiations. If \( \sim_a \) is the indistinguishability relation for agent \( a \), then we say that two histories \( h, h' \) are indistinguishable to agent \( a \), if \( h \equiv_a h' \).

For \( A \subseteq \text{Ag} \), let \( \sim_A^* = (\cup_{a \in A} \sim_a)^* \), where \( * \) denotes the reflexive transitive closure (wrt. composition relations), and its extension to histories is denoted \( \equiv_A^* \). For \( A \subseteq \text{Ag} \), let \( \sim_A^2 = \cap_{a \in A} \sim_a \), and its extension to histories is denoted \( \equiv_A^2 \).

**Strategies.** A deterministic memoryless strategy, or simply a strategy, for agent \( a \) is a function \( \sigma_a : \text{hist}(S) \rightarrow \text{Act}_a \). A strategy \( \sigma_a \) is uniform if for all \( h \equiv_a h' \), we have \( \sigma(h) = \sigma(h') \). The set of uniform strategies is denoted \( \Sigma(S) \). All strategies in the rest of the paper are uniform (although sometimes we will stress this fact and write “uniform strategy”).

For \( A \subseteq \text{Ag} \), let \( \sigma_A : A \rightarrow \Sigma(S) \) denote a function that associates a uniform strategy \( \sigma_a \) with each agent \( a \in A \). We write \( \sigma_A(a) = \sigma_a \), and call \( \sigma_A \) a joint strategy.

For \( h \in \text{hist}(S) \) write \( \text{out}(S, h, \sigma_A) \), called the outcomes of \( \sigma_A \) from \( h \), for the set of computations \( \pi \in \text{comp}(S, h) \) such that \( \pi \) is consistent with \( \sigma_A \), that is, \( \pi \in \text{out}(S, h, \sigma_A) \) iff (i) \( \pi_{|\pi|} = h \); (ii) for every position \( i \geq |h| \), there exists a joint-action \( J_i \in \text{ACT} \) such that \( \pi_{i+1} \in \delta(\pi_i, J_i) \), and for every \( a \in A, J_i(a) = \sigma_A(a)(\pi_{i+1}) \). We may drop \( S \) and write simply \( \text{out}(h, \sigma_A) \). Notice that, if \( A = \emptyset \), then \( \text{out}(h, \sigma_A) \) is the set of all paths starting with \( h \) (this is because \( \sigma_A \) is the empty function and (ii) above places no additional restrictions on the computations).

### 2.2 The Logic \( \text{ATL}_{\text{UTD}} \)

In this section we define the logic \( \text{ATL}_{\text{UTD}} \). Its syntax has been called \( \text{ATEL}^* \) (cf. [14]), and we interpret it on iCGS with history-based semantics and imperfect information.

**Syntax.** Fix a finite set of atomic propositions (atoms) \( \text{AP} \) and a finite set of agents \( \text{Ag} \). The history (\( \varphi \)) and path (\( \psi \)) formulas over \( \text{AP} \) and \( \text{Ag} \) are built using the following grammar:

\[
\begin{align*}
\varphi &::= p \mid \neg \varphi \mid \varphi \land \varphi \mid K_a \varphi \mid C_a \varphi \mid D_a \varphi \mid \langle A \rangle \psi \\
\psi &::= \varphi \mid \neg \psi \mid \varphi \land \psi \mid X \psi \mid \psi U \psi
\end{align*}
\]

where \( p \in \text{AP}, a \in \text{Ag}, \) and \( \text{Ag} \subseteq \text{Ag} \).

The class of \( \text{ATL}_{\text{UTD}} \) formulas is the set of history formulas generated by the grammar. The temporal operators are \( X \) (read “next”) and \( U \) (read “until”). The strategy quantifier is \( \langle A \rangle \) (“the agents in \( A \) can enforce \( \psi \))”, and the epistemic operators are \( K_a \) (“agent \( a \) knows that”), \( C_a \) (“it is common-knowledge amongst \( A \) that”), and \( D_a \) (“the agents in \( A \) distributively know that”).

**Semantics.** Fix an iCGS \( S \). We simultaneously define, by induction on the formulas, \( (S, h) \models \varphi \) where \( h \in \text{hist}(S) \) and \( \varphi \) is a history formula, and \( (S, \pi, m) \models \psi \) where \( \pi \in \text{comp}(S), m \geq 0 \), and \( \psi \) is a path formula:

\[
\begin{align*}
(S, h) \models \varphi \quad & \text{iff last}(h) \in \lambda(p), \text{ for } p \in \text{AP} \quad \text{ \text{(i)}} \\
(S, h) \models \neg \varphi \quad & \text{iff } (S, h) \not\models \varphi \quad \text{ \text{(ii)}} \\
(S, h) \models \varphi_1 \land \varphi_2 \quad & \text{iff } (S, h) \models \varphi_i, \text{ for } i \in \{1, 2\} \quad \text{ \text{(iii)}} \\
(S, h) \models \langle A \rangle \psi \quad & \text{iff for some joint strategy } \sigma_A \in \Sigma(S), \quad (S, \pi, |\pi| - 1) \models \psi \quad \text{ for all } \pi \in \text{out}(h, \sigma_A) \quad \text{ \text{(iv)}} \\
(S, h) \models K_a \varphi \quad & \text{iff for every history } h' \in \text{hist}(S), \quad h' \equiv_a h \text{ implies } (S, h') \models \varphi \quad \text{ \text{(v)}} \\
(S, h) \models C_a \varphi \quad & \text{iff for every history } h' \in \text{hist}(S), \quad h' \equiv_a h \text{ implies } (S, h') \models \varphi \quad \text{ \text{(vi)}} \\
(S, h) \models D_a \varphi \quad & \text{iff for every history } h' \in \text{hist}(S), \quad h' \equiv_a h \text{ implies } (S, h') \models \varphi \quad \text{ \text{(vii)}}
\end{align*}
\]

For a history formula \( \varphi \), write \( S \models \varphi \) to mean that \( (S, s) \models \varphi \) for every \( s \in S_0 \).

We isolate some important fragments.

1. The fragment \( \text{ATL}_{\text{UTD}} \) consists of history formulas \( \varphi \) defined by the grammar above, except with the following path formulas: \( \psi := X \varphi \mid \psi U \varphi \)

2. The fragment \( \text{ATL}^* \) consists of formulas of \( \text{ATL}_{\text{UTD}} \) (resp. \( \text{ATL}^* \)) that do not mention epistemic operators.

3. The CTL operator \( E \) (resp. \( A \)) is definable in \( \text{ATL}^* \) by \( \langle [0] \rangle \) (resp. \( \langle [0] \rangle \)). In particular, \( \text{ATL}_{\text{UTD}} \) is a syntactic fragment of \( \text{CTL}^* \). The fragment of \( \text{CTL}_{\text{UTD}} \) consisting of formulas of the form \( A \psi \), where \( \psi \) is a path formula, is denoted \( \text{TLT}_{\text{UTD}} \). Finally, \( \text{TLT} \) is the fragment of \( \text{TLT}_{\text{UTD}} \) that does not mention epistemic operators.

**Remark 1.** The definition of the semantics of \( \langle A \rangle \psi \) is the “objective” semantics of \( \langle A \rangle \), and captures the idea that a designer is reasoning about the existence of strategies. On the other hand, “subjective” semantics capture the idea that agents themselves are reasoning about the existence of strategies [31]. In Section 3.1 we define subjective semantics and extend our main result to deal with these.

**Model Checking.** We state the main decision problem of this work.

**Definition 4 (Model Checking).** Let \( C \) be a class of \( iCGS \) and \( F \) a sublanguage of \( \text{ATL}_{\text{UTD}} \). Model checking \( C \) against \( F \) specifications is the following decision problem: given \( S \in C \) and \( \varphi \in F \) as input, decide whether \( S \models \varphi \).

Model checking is undecidable in general. Actually, it is undecidable even if \( C \) consists of all \( iCGS \) with \( |\text{Ag}| = 3 \) and \( F \) consists of the single formula \( \langle [1, 2] \rangle G p \), see [6]. In Section 3 we prove that model checking PA-\( iCGS \) against \( \text{ATL}_{\text{UTD}} \) specifications is decidable.

**2.3 Examples**

We illustrate this definition with two scenarios: the epistemic puzzle of the muddy children [8], and a generalisation of the game Battleships to multiple players. We model (or sketch) the scenario as an \( iCGS \) only having public actions, and supply representative formulas of \( \text{ATL}_{\text{UTD}} \).

**Muddy Children.** We express this classic puzzle, as presented, e.g., in [8], in slightly different terms. There are \( n \) children, \( k \) of them with mud on their foreheads. Each child can see the forehead (and thus the muddy state) of all the other children, but not their own. Then the father enters the scene. The father can see the foreheads of all the muddy children. Each person can only make truthful statements about what they know.

The classic question is: what one statement can the father make that will lead each muddy child to learn that she
is indeed muddy. The answer is that the father declares, assuming that $k \geq 1$, that “at least one of you is muddy”.

There are a number of modeling decisions one can make to formalise this scenario, e.g., we must decide on the exact set of actions. Let $Ag = \{ F \} \cup \{ 1, 2, \ldots, n \}$. We give the father two actions, i.e., $Act_v = \{ 2m, \neg 2m \}$. We give the $i$th child two actions, i.e., $Act_i = \{ \text{know muddy}, \neg \text{know muddy} \}$.

The set of states is $S = \{ 0, 1 \}^n \times (ACT \cup \{ \epsilon \})$. If the current state is $(v, J)$ then $v_i = 1$ (resp. $v_i = 0$) means that the $i$th child is (resp. is not) muddy, and $J$ is the most recent joint action (i.e., means that no joint action has yet taken place).

The set of initial states is $S_0 = \{ 0, 1 \}^n \times \{ \epsilon \}$. The transition relation simply updates the last joint action: $\delta((v, J), J') = (v, J')$. The father is perfectly informed so $\sim_F = \{(s, s') : s \in S\}$, and each child only does not see herself, so $\sim_i = \{(v, J), (v', J') : \lambda_{J,j} = v_j, J = J'\}$. Finally, the atoms are $AP = \{m_i : i \leq n\}$, and $\alpha_a = \alpha \in Act_a, a \in Ag$. Finally, define $\lambda(m_i) = \{(s, J) : s_i = 1\}$ and $\lambda(\alpha_a) = \{(s, J) : J(a) = \alpha\}$.

So $m_i$ means that the $i$th child is muddy, and $\alpha_a$ means that agent $a$’s last action was $\alpha \in Act_a$. It is straightforward to check that we have defined an iCGS only having public actions.

Define the shorthand formula $Kw_a\phi$, read “agent $a$ knows whether $\phi$”, by the formula $K_a\phi \lor \neg K_a\neg \phi$. Consider the formula:

$$\langle\{ f, 1, \ldots, n\}\rangle [\bigwedge_{i=0}^n (G(a_i) \rightarrow \tilde{a}_i)] \land F(\bigwedge_{i<n} Kw_im_i)$$

where the first conjunction is over $a \in Ag, \alpha \in Act_a$, and $\tilde{a}_i$ is a formula representing the intended meaning of $a_i$, e.g., if $a_i = \text{know muddy}$ then $\tilde{a}_i = Kw_i m_i$. This expresses that all the agents have a truthful strategy so that eventually each child will know whether or not she is muddy.

The reader might wonder what would happen if we don’t explicitly express that the actions are truthful. Consider the following formula that says that the father has a strategy such that eventually the children know their muddy state:

$$\langle\{ f\}\rangle F \bigwedge_{i<n} Kw_im_i$$

This formula is true. Indeed, a strategy for the father is to play action $\exists m$ at the $i$th step iff the $i$th child is muddy.

In the next scenario, what agents see changes over time (unlike in the muddy children’s scenario). In both scenarios, what agents hear gets updated over time.

**Battleships.** We consider a game of battleships with three players. Each player has a 10 x 10-board with numeric coordinates from bottom-left (0,0) to top-right (9,9). Initially, each player can only see her own board. On her board each player displays her battleships: one carrier of size 5, two battleships of size 4, three cruisers of size 3, four submarines also of size 3, and five destroyers of size 2. We assume that ships are displayed either horizontally or vertically. As is standard, overlapping is not allowed.

Here we consider a synchronous version of battleships, structured in 2-phase rounds, where in phase 1 every player broadcasts the name $p$ of a player and cell $(n, m)$ of $p$’s board to attack. Then, in phase 2, the players truthfully state whether they have been hit or missed and the boards are updated accordingly. A player is eliminated once all her ships have been destroyed. The last player standing, if there is one, wins the game. The relevant atoms in this game are $\text{win}_i$, and $\text{lose}_i$ which state whether player $i$ has won or lost.

It is a routine exercise to build an iCGS only having public actions from this description. We only mention that the assumption that players are truthful can be built in to the iCGS, i.e., by limiting the available actions a player has in state 2.

Now consider the formula:

$$\langle\{ 1, 2\}\rangle F \langle\{ 1, 3\}\rangle F (\text{lose}_2 \lor \langle\{ 1\}\rangle F \text{win}_1)$$

This expresses that player 1 can collude with each of her enemies, in order to weaken player 2 or win the game.

### 3. DECIDABILITY OF PA-iCGS

In this section we prove that model checking PA-iCGS against $ATL^*_K,\Sigma,\text{D}$ specifications is decidable. This should be contrast with the fact that model checking arbitrary iCGS against $ATL$ specifications is undecidable [6].

**THEOREM 1.** Model checking PA-iCGS against $ATL^*_K,\Sigma,\text{D}$ specifications is $2\text{exptime}$-complete.

The bulk of this section is devoted to proving decidability. We then establish the complexity, and discuss further extensions of the result. Before giving the proof, we introduce an encoding $\mu$ of histories.

**DEFINITION 5.** Let $S$ be an iCGS. Let $\mu : S_0 \times \text{ACT}^* \rightarrow \text{hist}(S)$ denote the function mapping $(s_0, u)$ to the history starting at the initial state $s_0$ that results from the sequence of joint actions $u \in \text{ACT}^*$. That is, $\mu(s_0, u) = \text{hist}$ such that $h_0 = s_0, h_j = \delta(h_{j-1}, u_{j-1})$ for $1 \leq j \leq |u|$.

For PA-iCGS, the encoding is actually a bijection:

**REMARK 2.** Let $S$ be a PA-iCGS. Since each $\sim_a$ is reflexive, $\delta(s, \cdot) : \text{ACT} \rightarrow S$ is injective for every $s \in S$. Thus, $\mu : S_0 \times \text{ACT}^* \rightarrow \text{hist}(S)$ is a bijection. In particular, for every $h \in \text{hist}(S)$ and $s \in S_0$ there exists a unique $u \in \text{ACT}^*$ such that $\mu(s, u) = h$. This bijection allows us to encode histories of $S$ by (unique) elements of $S_0 \times \text{ACT}^*$.

An immediate consequence of having only public actions, but one that forms the foundation of our decidability proof, is that the moment different joint actions are taken, two histories become distinguishable.

**LEMMA 1.** Let $S$ be a PA-iCGS. For all $a \in Ag, u, u' \in \text{ACT}^*$ and $s, s' \in S_0$, if $\mu(s, u) \equiv_a \mu(s', u')$ then $u = u'$.

**PROOF.** If $\mu(s, u) \equiv_a \mu(s', u')$ then $|u| = |u'|$ and, for all $0 \leq j \leq |u|$, $\mu(s, u)_j \sim_a \mu(s', u')_j$. By the definition of having only public actions, $u_j = u'_j$ for all $j < |u|$. □

We prove Theorem 1 in the rest of this section. We use an automata-based marking algorithm. Such algorithms have been successfully applied to a number of logics, including $\text{CTL}^*$ [19] and $\text{ATL}^*$ [1] in the perfect information setting. **Automata theory.** Since our proof uses an automata-theoretic approach we now fix notations of word and tree automata. We remark that we only make use of standard properties of automata operating on finite words, infinite words, and infinite trees [20].

A deterministic finite-word automaton (DFW) is a tuple $M = (\Sigma, S, s_0, \rho, F)$ where $\Sigma$ is the input alphabet, $S$ is the finite set of states, $s_0 \in S$ the initial state, $\rho : S \times \Sigma \rightarrow S$
the deterministic transition function, and $F \subseteq S$ the set of final states. The run of $M$ on $u \in \Sigma^*$ is the finite sequence $s_0a_1 \cdots a_{|u|}$ where $p(s_i, a_i) = s_{i+1}$ for all $i < |u|$. The automaton accepts a word $u \in \Sigma^*$ iff the run of $M$ on $u$ ends in a final state. A DFWM is empty if it accepts no word. A set of strings $X \subseteq \Sigma^*$ is called regular if there is a DFWM $M$ that accepts $u \in \Sigma^*$ iff $u \in X$.

We also make use of automata operating on infinite words $\alpha \in \Sigma^\omega$: a deterministic parity word automaton (DPWM) is a tuple $T = (\Sigma, D, S, \rho)$ where all components are as for a DFWM except that $D$ is the finite set of directions, and $\rho : S \times \Sigma \rightarrow S^D$. The automaton operates on $\Sigma$-labelled $D$-ary branching trees, i.e., functions $f : T^D \rightarrow T$. A branch of $t$ is an infinite sequence $\beta \in D^\omega$. The run of $T$ in input $t$ is the $Q$-labeled $D$-ary branching tree $g : T^D \rightarrow Q$ such that $g(\epsilon) = s_0$ and $g(ud) = \mu(g(u), t(u))$. The automaton accepts the tree $f$ iff for every $\beta \in D^\omega$ (called a branch), the smallest color $k$ for which there are infinitely many $i$ with $c(s_i) = k$ is even (where $s_0s_1 \cdots$ is the run of $M$ on $\alpha$).

We also make use of automata operating on trees. A deterministic parity tree automaton (DPWT) is a tuple $T = (\Sigma, D, S, \rho, c)$ where all components are as for a DFWM except that $D$ is the finite set of directions, and $\rho : S \times \Sigma \rightarrow S^D$. The automaton operates on $\Sigma$-labelled $D$-ary branching trees, i.e., functions $f : T^D \rightarrow T$. A branch of $t$ is an infinite sequence $\beta \in D^\omega$. The run of $T$ in input $t$ is the $Q$-labeled $D$-ary branching tree $g : T^D \rightarrow Q$ such that $g(\epsilon) = s_0$ and $g(ud) = \rho(g(u), t(u))$. The automaton accepts the tree $f$ iff for every $\beta \in D^\omega$ (called a branch), the smallest color $k$ for which there are infinitely many $i$ with $c(s_i) = k$ is even.

The classes of DFWM and DPWT are effectively closed under the Boolean operations (complementation and intersection). Also, DFWM, DPWT and DPT can be effectively tested for emptiness. Finally, we make use of the following important fact connecting linear temporal logic with automata:

\begin{proposition} \cite{35,29}. Every LTL formula $\psi$ over atoms AP can be effectively converted into a DFWM $P_\psi$ with input alphabet $2^{AP}$ that accepts a word $\alpha \in 2^{AP}$ iff $\alpha \models \psi$. Moreover, the DFWM has double-exponentially many states and single-exponentially many colours. \end{proposition}

\begin{proof} outline. \end{proof}

We proceed by induction on the formula $\varphi$ to be checked. We build a DFWM that accepts all encodings of histories $h$ such that $(S, h) \models \varphi$. Precisely, we build a DFWM $M^\varphi$ that accepts a sequence of joint actions $a \in ACT^*$ iff $(S, \mu(s, u)) \models \varphi$. The atomic case is immediate, and the Boolean cases follow from the effective closure of DFWM under complementation and intersection. The $\varphi = \exists a \varphi'$ case is done by simulating the DFWM $M^\varphi_a$ for $t \in S_0$, and recording whether or not $\mu(s, u) \models \mu(t, u)$; the other knowledge operators are similar. The strategic operator $\varphi = \langle A \rangle \psi$ is done as follows: first we show that we can assume $\psi$ is an LTL formula, and then we build a DFWM for the formula $\psi$: we build the DFWM that simulates the DFWM, and when the input ends we use a tree automaton to decide if there is a joint strategy that ensures that the DFWM accepts all computations consistent with that joint strategy.

\begin{generalisation} of the labeling function. \end{generalisation}

We first generalise the labeling function of iCGS so that atoms are regular sets of histories (instead of state labelings).

\begin{definition} A generalised iCGS is a tuple $S = (Ag, AP, \{\ACT_a\}_{a \in Ag}, S, S_0, 0, A, \{\sim_a\}_{a \in Ag}, \Lambda)$ where all entries are as for iCGS, except that $\Lambda : AP \rightarrow 2^S$ is replaced by a function $\Lambda : AP \rightarrow 2^{\text{hist}(S)}$ such that $\Lambda(p) \subseteq \text{hist}(S)$ is a regular set of histories, i.e., there exists a DFWM over the alphabet $S$ accepting $h \in \text{hist}(S)$ iff $h \in \Lambda(p)$.

Then, we redefine the atomic case of the semantics of ATL^\text{C,E} : (S, h) \models p \text{ iff } h \in \Lambda(p)$. It is immediate that generalised iCGS are indeed more general than iCGS:

\begin{lemma} \end{lemma}

\begin{proof} First, note that $\Lambda(p)$ is regular since a DFWM can read the history $h$ and store in its state whether or not the last state it read is in $\Lambda(p)$ or not. Second, the fact that $S \models \varphi$ iff $S' \models \varphi$ holds by a straightforward induction on the structure of $\varphi$. \end{proof}

A generalised PA-iCGS is a generalised iCGS that only has public actions, i.e., it satisfies the conditions in Definition 3 (which does not depend on the labeling). For the rest of the proof we view $S$ as a generalised PA-iCGS.

\begin{inductive} \end{inductive}

For every history formula $\varphi$ and initial state $s \in S_0$ we will build a DFWM $M^\varphi_s$ such that for every $u \in ACT^*$,

$$M^\varphi_s \text{ accepts } u \text{ iff } (S, \mu(s, u)) \models \varphi.$$ 

From this it is easy to get the decidability stated in the theorem: simply check that $\epsilon$ is accepted by every $M^\varphi_s$ with $s \in S_0$.

We build the DFWM $M^\varphi_s$ simultaneously for all $s \in S_0$, by induction on $\varphi$.

\begin{varphi} is an atom. \end{varphi}

Say $\varphi = p \in AP$ and let $s \in S_0$. The required DFWM $M^\varphi_s$ should accept $u \in ACT^*$ iff $\mu(s, u)$ is accepted by the DFWM $\Lambda(p)$. To do this we define $M^\varphi_s$ to simulate $S$ and the DFWM $R = (S, Q, q_0, F)$ for $\Lambda(p)$ in parallel, i.e., by taking a product of $S$ and $R$. Formally, define $M^\varphi_s = (ACT, S \times Q, (t, q_0, \tau, F'))$ where the transition function $\tau$ maps state $(s,q)$ and input $a \in ACT$ to state $(\delta(s,a), \rho(q,s))$, and the final states $F'$ are of the form $(s,q)$ where $\rho(q,s) \in F$.

\begin{varphi} is a Boolean combination. \end{varphi}

Let $s \in S_0$. The Boolean combinations follow from the effective closure of DFWM under complementation and intersection. Indeed, $M^\varphi_s$ is formed by complementing the final states of $M^\varphi_s$ and $M^\varphi_{s' \cdot \neg \varphi}$ is the product of the $M^\varphi_s$. $\varphi$ is of the form $\exists a \varphi'$. Let $s \in S_0$. By induction, we have DFWM $M^\varphi_s$ for $t \in S_0$. The required DFWM should accept a string $u$ iff, for every $t \in S_0$, if $\mu(s, u) \equiv \mu(t, u)$ then $M^\varphi_t$ accepts $u$.

To do this, the DFWM will simulate, in parallel, each $M^\varphi_t$ for $t \in S_0$. This is done by forming their product, i.e., the states of the product are $q : S_0 \rightarrow Q$ where $Q$ is the union of the state sets of the $M^\varphi_t$ for $t \in S_0$, and there is a transition in the product from $q$ to $q'$ on input $J \in ACT$ if for each $t \in S_0$ there is a transition in $M^\varphi_t$ from $q(t)$ to $q'(t)$ on input $J$. Instrument this product by recording, on input $u \in ACT^*$ a function $f_u : S_0 \rightarrow S$ and a set $G_u \subseteq S_0$ with the following properties:

- $f_u(t) = \langle \mu(t, u) \rangle$
- $t \in G_u$ iff for every prefix $v$ of $u$, $f_v(s) \sim f_u(t)$ (thus, initially $G_u = \{ t \in S_0 : t \sim_s \}$, and the moment $f_u(s) \not\sim f_u(t)$ we remove $t$ from $G_u$).
A state \((f, G, q)\) is final if, for every \(t \in G\) it is also the case that \(q(t)\) is a final state of \(M_f^s\).

\(\varphi\) is of the form \(\exists \sigma \varphi'\). This is identical to the \(K_\sigma\) case except replace \(\leadsto_\varnothing\) by \(\Delta^{\varnothing}_\sigma\) and replace \(\varnothing_0\) by \(\Xi_\sigma^0\).

\(\varphi\) is of the form \(\forall \sigma \varphi\). This is identical to the \(K_\sigma\) case except replace \(\leadsto_\varnothing\) by \(\Delta^{\varnothing}_\sigma\) and replace \(\varnothing_0\) by \(\Xi_\sigma^0\).

\(\varphi\) is of the form \(\langle \overline{\varphi} \rangle \psi\). We proceed in two steps. First, we show how to linearise path formulas, and then we show how to encode strategies as trees so that we can build the promised DFV.

**Linearising path formulas.** One can think of an ATL\(\ast,\mathbb{C},\mathbb{D}\) path formula \(\psi\) as an ATL formula \(\text{lin}(\psi)\) over a fresh set of atoms \(\text{max}(\psi)\), the maximal history subformulas of \(\psi\). This translation of ATL\(\ast,\mathbb{C},\mathbb{D}\) path formulas to ATL formulas does not make use of the assumption that the \(\mathbb{C}\)S \(\mathbb{D}\) only has public actions, and it is analogous to the translation of ATL\(\ast\) (or CTL\(\ast\)) path formulas to ATL formulas over maximal state subformulas [19, 1].

We briefly discuss the translation. Let \(\text{lin}(\psi)\) be the set of history subformulas of \(\psi\) that are maximal, i.e., a history formula \(\varphi \in \text{lin}(\psi)\) if it occurs in \(\psi\) and that occurrence is not a subformula of any other occurrence of a history subformula of \(\psi\). If \(\psi\) is a path formula, define \(\text{lin}(\psi)\) to be the ATL formula where for each \(\varphi \in \text{max}(\psi)\) there is a fresh atom \(\overline{\varphi}\) such that every occurrence of \(\varphi\) in \(\psi\) is replaced by \(\overline{\varphi}\). For example, consider \(\psi = (p \lor \langle \overline{\varphi} \rangle K_{\sigma}p) \leftrightarrow X \neg C_{\mathbb{D}}p\). The history subformulas of \(\psi\) are \(\{p, \langle \overline{\varphi} \rangle K_{\sigma}p, \neg C_{\mathbb{D}}p, C_{\mathbb{D}}p\}\). Then \(\text{max}(\psi) = \{p, \langle \overline{\varphi} \rangle K_{\sigma}p, \neg C_{\mathbb{D}}p\}\). Thus \(\text{lin}(\psi)\) is the ATL formula \(\overline{\varphi} \lor \langle \overline{\varphi} \rangle K_{\sigma}p \lor X \neg C_{\mathbb{D}}p\) over the atoms \(\overline{\varphi}\) for \(\varphi \in \text{max}(\psi)\). Since, by induction, we have built DFV for each boxed atom, for the remainder of the proof we assume that \(\psi\) is an ATL formula.

**Construction of \(M_{\langle \overline{\varphi} \rangle \psi}^s\) for an ATL formula \(\psi\).** We will build a DFV \(E_{\psi,s}^\ast\) over ACT that accepts \(\alpha \in \text{ACT}^\omega\) iff \((S, \mu(s, \alpha)) \models \psi\). The promised DFV \(M_{\langle \overline{\varphi} \rangle \psi}^s\) reads \(u \in \text{ACT}^\ast\) and simulates \(E_{\psi,s}^\ast\). Suppose after reading \(u\) the state of \(E_{\psi,s}^\ast\) is \(q\). Then \(M_{\langle \overline{\varphi} \rangle \psi}^s\) accepts, i.e., \(q\) is defined to be a final state of \(M_{\langle \overline{\varphi} \rangle \psi}^s\) iff there exists uniform \(\sigma\) such that for every \(\alpha \in \text{ACT}^\ast\), if \(\mu(s, u, \alpha) \in \text{out}(S, \mu(s, u), \sigma)\) then \(\sigma\) is accepted by \(E_{\psi,s}^\ast\) starting from state \(q\) and (thus \((S, \mu(s, u)) \models \langle \overline{\varphi} \rangle \psi\), as required). These latter realisability problems (one for each \(\overline{\varphi}\)) are solved offline by constructing DFV \(F_{\psi,s,q}\) and testing them for non-emptiness. We now show how to build the automata \(E_{\psi,s}^\ast\) and \(F_{\psi,s,q}\).

**Construction of DFV \(E_{\psi,s}^\ast\).** To build \(E_{\psi,s}^\ast\), begin by writing \(A\text{P}(u)\) for the finite set of atoms appearing in \(\psi\). First, for each \(p \in A\text{P}(u)\), let \(D^p\) be a DFV for the regular set \(\{u \in \text{ACT}^\ast : \mu(s, u) \in \Lambda(p)\}\). Second, by Proposition 1, every ATL formula \(\psi\) can be converted into a DFV \(D_\psi\) over alphabet \(2^{A\text{P}(u)}\) that accepts all word models of that formula. Let \(Q_\psi\) be the states and \(\Delta_\psi : Q_\psi \times 2^{A\text{P}(u)} \to Q_\psi\) the transition of the DFV. Now, the DFV \(E_{\psi,s}^\ast\) simulates \(D_\psi\) and each DFV \(D^p\). The automaton does this by storing and updating a state \(q_u \in Q_u\) and a function \(f_u\) such that \(f_u(p)\) is the state of \(D^p\) (i.e., \(u : A\text{P}(u) \to Q_u\) where \(Q_u\) is the union of states of the \(D^p\)\s). Transitions of \(E_{\psi,s}^\ast\) are as follows: from state \((q_u, f_u)\) and input \(d \in \text{ACT}\) the next state \((q_{ud}, f_{ud})\) satisfies that \(q_{ud} = \Delta_\psi(q_u, Z)\) where \(p \in Z\) iff \(f_u(p)\) is a final state of \(D^p\), and \(f_{ud}(p)\) is the state resulting from applying

\[\text{Note that although } p \text{ has a non-maximal occurrence in } \psi, \text{ it is included in max}(\psi) \text{ since it has at least one occurrence which is maximal, i.e., on the left-side of } U.\]
1. Atomic: $\|p\| = O(1)$ for $p \in AP$.

2. Negation: $\|\neg \varphi\| = \|\varphi\|$.

3. Conjunction: $\|\varphi \land \varphi'\| = \|\varphi\| \times \|\varphi'\|$. 

4. Epistemic: $\|K_\alpha \varphi\| = (2 \times |S| \times \|\varphi\|)^{|S|}$, and the same for $\|C_A \varphi\|$ and $\|D_A \varphi\|$. 

5. Strategic: $\|\langle A \rangle \psi\| = 2^{O(|\text{lin}(\psi)|)} \times \|\tilde{\psi}\| \times |\text{AP}(\text{lin}(\psi))| \times O(\Sigma^2)$ if $\text{lin} \psi$ is the linearisation of $\psi$, $\tilde{\psi}$ is the largest history subformula of $\psi$, and $\text{AP}(\cdot)$ is the set of atomic predicates occurring in its argument.

The last case requires some explanation. The DPT accepting the set of all joint-strategies $\sigma_A$ has size $O(2^{|S|^2})$, and has two colours. The DFW $D_F$ has double-exponentially many states and single-exponentially many colours (in the size of $\text{lin} \psi$) [35, 29]. The DFW $D_{\psi, a, \theta} \times |\psi|$. For the strategy case we incur a cost to solve the strategy below, i.e. - solving the emptiness of the DPT $D_{\psi, a, \theta}$. The cost of solving the emptiness of a DPT with $n$ states and $m$ colours is at most $n^{O(m)}$ [16]. This, the time for constructing $M^*_{\psi}$ is at most $n^{O(m)}$ where $n = 2^{\Omega(|\text{lin}(\psi)|)} \times |\tilde{\psi}| \times |\text{AP}(\text{lin}(\psi))| \times O(\Sigma^2)$ and $m = 2^{O(|\text{lin}(\psi)|)}$.

Finally, let $\varepsilon = |S| + |\varphi|$. The time for constructing $M^*_{\psi}$ of each step of the procedure can be bounded above by $2^{O(|\varepsilon|^2)} \times |\varepsilon|^{O(|\varepsilon|)}$ where $|A|$ is the size of the largest DFW representing atoms of the generalised PA-ICGS (a maximum exists since AP is finite). Since there are at most $\varepsilon$ steps, the time, then, as the size of the resulting DFW is $2^{O(|\varepsilon|^2)}$. Testing if $\varepsilon$ is accepted by this automaton has no additional cost. Thus, our algorithm runs in $2\text{EXPTIME}$.

Lower-Bound. To prove the lower bound in Theorem 1 we reduce from a known $2\text{EXPTIME}$-hard problem, i.e., model checking a CGS with $Ag = \{a, b\}$ against a formula of the form $\langle(a) \rangle \psi$ for an ATL formula $\psi$ [27, 28]. One can translate a two-player CGS $S$ into a polynomially larger CGS $S'$ with public actions such that $S \models \varphi$ if $S' \models \varphi$. Indeed, the agents, actions, and, atoms are the same, $S' = S \times (\text{ACT} \cup \{a\})$, $S'_0 = S_0 \times \{a\}$, $\delta'((s, d), a') = \delta(s, d', a')$, and $\lambda'(p) = \{s, d : s \in \lambda(p)\}$ (note that $S$ has two players, $|\text{ACT}| = |\text{Act}_1| \times |\text{Act}_2|$), and thus $|S'|$ is polynomial in the size of $S$).

3.2 Subjective Semantics

In this section we show that our result still holds if we use subjective semantics instead of objective semantics.

Our definition of semantics of $\langle A \rangle \psi$ is called “objective” (see Remark 1). An alternative definition is called “subjective”: replace “for all $\pi \in \text{out}(h, \tau)$” by “for all $\pi \in \bigcup_{a \in A, h' : \sigma_A(h', \sigma_A)} \text{out}(S', h')$” in the semantics $(S, h) \models \langle A \rangle \psi$. Subjective semantics expresses, intuitively, that the agents $A$ know that a given strategy will guarantee a certain outcome. We remark that in this case $K_a \varphi$ is definable in terms of $\langle A \rangle \varphi$, i.e., $\langle a \rangle \varphi \cup \varphi$.

The decidability proof of Theorem 1 is for objective semantics. To deal with subjective semantics proceed as follows. For $u \in \text{ACT}^*$ let $T_u \subseteq S_0$ be the set of $t$ such that there exists $a \in A$ with $t \equiv_a \pi$. The DFW $M^*_{\psi, a, \theta}$ simulates $E_{\psi, t}$ for all $t \in T_u$. After reading $u$, each automaton $E_{\psi, t}$ for $t \in T_u$ in some state, say $q_t$. The DFW is required to accept $u$ iff there exists a joint strategy $\sigma_A$ such that for every $a \in \text{ACT}^*$ and $t \in T_u$, if $\mu(t, u \cdot a) \in \text{out}(S, \mu(t, u), \sigma_A)$ then $a$ is accepted by $E_{\psi, t}$ starting from $q_t$. In order to decide the right-hand side we build the DPT $F_{\psi, t, q_t}$ for $t \in T_u$ (as we did above for $s$). Then we check if the intersection of the automata $F_{\psi, t, q_t}$ for $t \in T_u$ is non-empty.

4. COMPARISON WITH BROADCAST ENVIRONMENTS

The work most closely related to ours is [33] in which the authors show that one can decide if a given formula of knowledge and linear-time, which we denote LK, is realisable (by a tuple of uniform strategies) assuming that the environment is a “broadcast environment”. In this section we denote the logic from [33] by LK.

The relationship with [33] is twofold: i) the realisability problem for LK can be reduced to model checking $\text{ATL}^*$ specifications, and ii) our notion of “having only public actions” (Definition 3) is orthogonal to “broadcast environments”. We now supply the justifications for i) and ii).

i) LK realisability can be reduced to model-checking $\text{ATL}^*$. We show how to reduce the realisability problem for LK to the model checking problem for $\text{ATL}^*$. To do so, we first recall the syntax and semantics of LK from [33]. The syntax is defined as the set of formulas $\psi$ generated by the following grammar:

$$\psi ::= p \mid \neg \psi \mid \psi \land \psi \mid X \psi \mid \psi U \psi \mid K_a \psi$$

where $p \in AP$, $a \in Ag$, and $A \subseteq Ag$ is non-empty.

The semantics $\models_{LK}$ is defined over $(S, \pi, m)$ where $S$ is an iCGS, $\pi \in \text{comp}(S)$ and $m \in N$. We denote the satisfaction relation $\models_{LK}$ to distinguish it from $\models$. The Boolean and temporal operators are as usual:

$$(\pi, m) \models_{LK} p \text{ if } \pi_m \in \lambda(p)$$

$$(\pi, m) \models_{LK} \neg \psi \text{ if } (\pi, m) \not\models_{LK} \psi$$

$$(\pi, m) \models_{LK} \psi_1 \land \psi_2 \text{ if } (\pi, m) \models_{LK} \psi_i \text{ for } i \in \{1, 2\}$$

$$(\pi, m) \models_{LK} X \psi \text{ if } (\pi, m, \pi_1) \models_{LK} \psi$$

$$(\pi, m) \models_{LK} \psi U \psi \text{ if for some } j \geq m, (\pi, j) \models_{LK} \psi_2, \text{ and for all } k \text{ with } m \leq k < j, (\pi, k) \models_{LK} \psi_1$$

The epistemic operator $K_a$ is follows:

$$(\pi, m) \models_{LK} K_a \psi \text{ if } (\pi', m') \models_{LK} \psi \text{ for all } \pi' \in \text{comp}(S) \text{ such that } \pi \equiv_m \pi' \equiv_m (\text{in particular, } m = m')$$

An LK-formula $\psi$ is realisable if there exists a uniform strategy $\sigma_A$ such that for all $a_0 \in S_0$ and all $\pi \in \text{out}(S, a_0)$, we have that $(S, \pi, 0) \models \psi$.

We now present the reduction. Let $\psi$ be a formula of LK. Define $\psi$, a path formula of CTL$^*$, by recursively replacing $K_a \psi$ by $K_a \psi$ and then replacing any $\psi$ with a path formula of $\text{ATL}^*$ as we did above for $\pi$. We claim that for all iCGS $S$, $\psi$ is realisable if $\psi$ is a path formula of $\text{CTL}^*$ and $\psi$ is a formula of $\text{ATL}^*$. To see this it is sufficient to establish the following inductive hypothesis: $$(S, \pi, m) \models_{LK} \psi \text{ if } (S, \pi, m) \models_{LK} \psi$$

To prove the inductive hypothesis use that the following are equivalent to $(S, \pi, m) \models_{LK} K_a \psi$: 

$$\psi ::= p \mid \neg \psi \mid \psi \land \psi \mid X \psi \mid \psi U \psi \mid K_a \psi$$
1. \((S, \pi', m) \models \psi\) for all \(\pi' \in \text{comp}(S)\) such that \(\pi' \leq m \equiv_a \pi \leq m\) (by the definition of \(\models\));

2. \((S, \pi', m) \models \psi\) for all \(\pi' \in \text{comp}(S)\) such that \(\pi' \leq m \equiv_a \pi \leq m\) (by the inductive hypothesis);

3. \((S, \pi, m) \models K_a A \psi\) (by definition of \(\models\)).

ii) Broadcast environments and PA-iCGS are incomparable. We briefly describe how [33] models “broadcast environment”. That work defines an interpreted systems (see [8] for background on these) in which each agent’s local state consists of a private part (that only depends on its local actions) and a shared part (that depends on the joint actions, but that is the same for all agents). In our terminology a broadcast environment is an iCGS with \(A_g = \{e, 1, 2, \ldots, n\}\) whose state set is of the form \(S = L_o \times \prod_{i \leq n} L_i\) (for some finite sets \(L_o\) for \(a \in A_g\)), and whose transition maps state \((l_o, l_1, \ldots, l_n)\) and joint-action \(J \in \text{ACT}\) to the state \((\tau_e(l_e, J), \tau_1(l_1, J(1)), \ldots, \tau_n(l_n, J(n)))\) where \(\tau_e : L_o \times \text{ACT} \rightarrow L_o\) and \(\tau_i : L_i \times \text{ACT}_i \rightarrow L_i\) are functions (similar to the evolution functions in interpreted systems), except that \(L_i\) for \(i \neq e\) does not depend on joint-actions, only on local actions). Finally, define \((l'_i, l'_1, \ldots, l'_n) \sim (l_i', l'_1, \ldots, l'_n)\) iff \(l_i = l'_i\) and \(F(l_i) = F(l'_i)\) where \(F : L_o \rightarrow O\) is a function mapping environment states to some fixed set \(O\) of observations (note that \(F\) and \(O\) are independent of \(e\), and thus all agents have the same observation of the environment).

Now, the set of PA-iCGS is incomparable (wrt. subset) with these iCGS. On the one hand, setting \(L_o = \text{ACT} = O\) and \(F\) to be the identity function, results in an iCGS having only public actions. On the other hand, we allow \(L_i\) to depend on the joint-actions (not just the local actions).

5. CONCLUSIONS

In this paper we put forward a class of CGS with imperfect information, namely the iCGS only having public actions, which admit a decidable model checking problem, even in the presence of perfect recall. This is in contrast with the fact that even realisability of safety properties on arbitrary iCGS is undecidable [6]. Specifically, we considered a rich formal language to express complex strategic and epistemic properties of agents in MAS. This is the extension ATL\(_{K,C,D}\) of the alternating-time temporal logic ATL\(^*\), with operators for individual, common, and distributed knowledge. We provided these languages with a semantics in terms of iCGS, according to both the objective and subjective interpretation of ATL modalities. Most importantly, we identified a subclass of iCGS – those having only public actions, or PA-iCGS – for which we were able to prove that the model-checking problem is decidable. The interest of these results lies in the fact that PA-iCGS capture many important MAS scenarios, including certain games of imperfect information, epistemic puzzles, blackboard systems, face to face communication, etc. Indeed, all scenarios mentioned in previous work on broadcast environments [23, 33] can be captured by PA-iCGS.

A number of extensions of ATL\(^*\) have been proposed in order to express classic solutions concepts (like Nash Equilibria) [12, 13, 24, 25]. The decidability of model checking PA-iCGS against epistemic extensions of these strategy logics is currently unexplored.

Notwithstanding their generality, there are many features of MAS that are not naturally expressed within PA-iCGS or broadcast environments. We discuss some of them:

Asynchronous recall. Social media like Twitter make use of public actions, but are more naturally modeled as asynchronous MAS (rather than synchronous systems, as we do).

Bounded-recall. Games like Bridge and Stratego are interesting to play in part because humans have to remember some of the history of a play, a feature that might be modeled by bounded recall (rather than perfect recall). However, restricting agents to finite-memory strategies also results in undecidability on arbitrary iCGS [36]. On the other hand, our proof implies that if a formula \(\langle A \rangle \psi\) is true then there are finite-memory strategies witnessing this fact, and if a formula \(K_a \varphi\) is true then there is a finite-state machine that accepts exactly the histories making \(K_a \varphi\) true. This suggests that our results can be used to model agents of bounded-recall.

Probabilities. Several scenarios, such as card games and security protocols, involve probability either at the level of the iCGS or at the level of strategies.

In future work we plan to investigate the points raised above, as well as to develop optimal model checking algorithms for fragments of ATL\(_{K,C,D}\) and to implement them in an extension of the MCMAS tool for MAS verification [22].

Acknowledgements. F. Belardinelli acknowledges the support of the ANR JCJC Project SVeDaS (ANR-16-CE40-0021). S. Rubin is a Marie Curie fellow of the Istituto Nazionale di Alta Matematica. The authors thank Benjamin Aminof for fruitful discussions.

REFERENCES


