# 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
DIETI
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, $\mathsf{LTL}_{\mathbb{K}}$, 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 $\mathsf{ATL}^*_{\mathbb{K},\mathbb{C},\mathbb{D}}$ on PA-iCGS is decidable, specifically 2EXPTIME-complete. Thus, the joint complexity of model-checking is the same as that of $\mathsf{ATL}^*$ with perfect information [1]. Morever, 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 $\mathsf{ATL}^*_{\mathbb{K},\mathbb{C},\mathbb{D}}$, 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, $\mathsf{CTL}^*$ 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 $\mathsf{LTL}_{\mathbb{K},\mathbb{C}}$ 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 $\mathsf{LTL}_{\mathbb{K}}$ 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

**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.*

non-elementary [5, 11]; and, model-checking ATL in which all coalitions operate with a single indistinguishability relation 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 observation or information sets of the agents; ii) on the the architectures 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 architectures 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 broadcast environments (which restrict the underlying iCGS) 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 strategic 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 Announcement 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 explictly 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 (iCGS) that only have public actions (PA-iCGS). The logic is $\mathsf{ATL}^*_{\mathbb{K},\mathbb{C},\mathbb{D}}$, an extension of Alternating Time Temporal Logic ($\mathsf{ATL}^*$) which includes strategic operators ($\langle\!\langle A \rangle\!\rangle$ for $A \subseteq Ag$) as well as epistemic operators for individual-knowledge ($\mathbb{K}_a$ for $a \in Ag$), common-knowledge ($\mathbb{C}_A$ for $A \subseteq Ag$), and distributed-knowledge ($\mathbb{D}_A$ for $A \subseteq Ag$).

**Notation**. For an infinite or non-empty finite sequence $u \in X^\omega \cup X^+$ write $u_i$ for the $i$th element of $u$, i.e., $u = u_0 u_1 \cdots$. The empty sequence is denoted $\epsilon$. The length of a finite sequence $u \in X^*$ is denoted $|u|$, its last (resp. first) element is denoted $last(u)$ (resp. $first(u)$). Note that $last(\epsilon) = first(\epsilon) = \epsilon$. For $i < |u|$ write $u_{\leq i}$ for the prefix $u_0 \cdots u_i$. For a vector $v \in \prod_i X_i$ we denote the $i$-th co-ordinate of $v$ by $v(i)$. In particular, for $F \in \prod_i (X_i)^Y$ we may write $F(i) \in X^Y$ and $F(i)(y) \in X_i$.

## 2.1 iCGS with only Public Actions

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

Definition 1 (iCGS). *An* imperfect information concurrent game structure (iCGS) *is a tuple*

$$\mathsf{S} = \langle Ag, AP, \{Act_a\}_{a \in Ag}, S, S_0, \delta, \{\sim_a\}_{a \in Ag}, \lambda \rangle$$

*where:*

- *$Ag$ is the finite non-empty set of* agent names*;*
- *$AP$ is the finite non-empty set of* atomic propositions*;*
- *$Act_a$ is the finite non-empty set of* actions *for $a \in Ag$;*
- *$S$ is the finite non-empty set of* states*;*
- *$S_0 \subseteq S$ is the non-empty set of* initial states*;*
- *$\delta : S \times \mathsf{ACT} \to S$ is the* transition function*, where the set $\mathsf{ACT}$ of* joint-actions *is the set of all functions $J : Ag \to \bigcup_a Act_a$ such that $J(a) \in Act_a$. The transition function assigns to every state $s$ and joint-action $J$, a successor state $\delta(s, J)$;*
- *$\sim_a \subseteq S^2$ is the* indistinguishability relation *for agent $a$, which is an equivalence relation; the equivalence class $[s]_a$ of $s \in S$ under $\sim_a$ is called the* observation set *of agent $a$;*
- *$\lambda : AP \to 2^S$ is the* labeling *function that assigns to each atom $p$ the set of states $\lambda(p)$ 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 $\sim_a = \{(s, s) : s \in S\}$ for all $a \in Ag$.*

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 iCGS that we call PA-iCGS.

Definition 3 (PA-iCGS). *An iCGS* only has public actions *if for every agent $a \in Ag$, states $s, s' \in S$, and joint actions $J, J' \in \mathsf{ACT}$, if $J \neq J'$ and $s \sim_a s'$ then $\delta(s, J) \not\sim_a \delta(s', J')$. 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 $T \times (\mathsf{ACT} \cup \{\epsilon\})$, where $\epsilon$ refers to the situation that no actions have yet been played, and ii) the indistinguishability relations $\sim_a$ satisfy that if $(t, J) \sim_a (t', J')$ 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 information**. A *path* in $\mathsf{S}$ is a non-empty infinite or finite sequence $\pi_0 \pi_1 \cdots \in S^\omega \cup S^+$ such that for all $i$ there exists a joint-action $J(i) \in \mathsf{ACT}$ such that $\pi_{i+1} \in \delta(\pi_i, J(i))$. Paths that start with initial states are called *histories* if they are finite and *computations* if they are infinite. The set of computations in $\mathsf{S}$ is written $\mathsf{comp}(\mathsf{S})$, and the set of computations in $\mathsf{S}$ that start with history $h$ is written $\mathsf{comp}(\mathsf{S}, h)$. We define $\mathsf{hist}(\mathsf{S})$ and $\mathsf{hist}(\mathsf{S}, h)$ similarly.

We use the following notation: if $\sim$ is a binary relation on $S$ we define the extension of $\sim$ to histories as the binary relation $\equiv$ on $\mathsf{hist}(\mathsf{S})$ define by $h \equiv h'$ iff $|h| = |h'|$ (i.e., synchronicity) and $h_j \sim h'_j$ for all $j \leq |h|$ (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 $A \subseteq Ag$, let $\sim_A^{\mathbb{C}} = (\cup_{a\in A} \sim_a)^*$, where $*$ denotes the reflexive transitive closure (wrt. composition of relations), and its extension to histories is denoted $\equiv_A^{\mathbb{C}}$. For $A \subseteq Ag$, let $\sim_A^{\mathbb{D}} = \cap_{a\in A} \sim_a$, and its extension to histories is denoted $\equiv_A^{\mathbb{D}}$.

**Strategies**. A *deterministic memoryfull strategy*, or simply a *strategy*, for agent $a$ is a function $\sigma_a : \mathsf{hist}(\mathsf{S}) \to Act_a$. 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 $\Sigma(\mathsf{S})$. All strategies in the rest of the paper are uniform (although sometimes we will stress this fact and write "uniform strategy").

For $A \subseteq Ag$, let $\sigma_A : A \to \Sigma(\mathsf{S})$ 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 $h \in \mathsf{hist}(\mathsf{S})$ write $\mathsf{out}(\mathsf{S}, h, \sigma_A)$, called the *outcomes of* $\sigma_A$ *from* $h$, for the set of computations $\pi \in \mathsf{comp}(\mathsf{S}, \mathsf{h})$ such that $\pi$ is consistent with $\sigma_A$, that is, $\pi \in \mathsf{out}(\mathsf{S}, h, \sigma_A)$ iff (i) $\pi_{\leq |h|-1} = h$; (ii) for every position $i \geq |h|$, there exists a joint-action $J_i \in \mathsf{ACT}$ such that $\pi_{i+1} \in \delta(\pi_i, J_i)$, and for every $a \in A$, $J_i(a) = \sigma_A(a)(\pi_{\leq i})$. We may drop $\mathsf{S}$ and write simply $\mathsf{out}(h, \sigma_A)$. Notice that, if $A = \emptyset$, then $\mathsf{out}(h, \sigma_A)$ 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 $\mathsf{ATL}^*_{\mathbb{K},\mathbb{C},\mathbb{D}}$

In this section we define the logic $\mathsf{ATL}^*_{\mathbb{K},\mathbb{C},\mathbb{D}}$. Its syntax has been called 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) AP* and a finite set of *agents Ag*. The *history ($\varphi$) and path ($\psi$) formulas over AP and Ag* are built using the following grammar:

$$\begin{array}{lcl} \varphi & ::= & p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbb{K}_a\varphi \mid \mathbb{C}_A\varphi \mid \mathbb{D}_A\varphi \mid \langle\!\langle A\rangle\!\rangle\psi \\ \psi & ::= & \varphi \mid \neg\psi \mid \psi \wedge \psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi \end{array}$$

where $p \in AP$, $a \in Ag$, and $A \subseteq Ag$.

The class of $\mathsf{ATL}^*_{\mathbb{K},\mathbb{C},\mathbb{D}}$ *formulas* is the set of history formulas generated by the grammar. The *temporal operators* are $\mathsf{X}$ (read "next") and $\mathsf{U}$ (read "until"). The *strategy quantifier* is $\langle\!\langle A\rangle\!\rangle$ ("the agents in $A$ can enforce $\psi$"), and the *epistemic operators* are $\mathbb{K}_a$ ("agent $a$ knows that"), $\mathbb{C}_A$ ("it is common-knowledge amongst $A$ that"), and $\mathbb{D}_A$ ("the agents in $A$ distributively know that").

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

$(\mathsf{S}, h) \models p$ iff $last(h) \in \lambda(p)$, for $p \in AP$.
$(\mathsf{S}, h) \models \neg\varphi$ iff $(\mathsf{S}, h) \not\models \varphi$.
$(\mathsf{S}, h) \models \varphi_1 \wedge \varphi_2$ iff $(\mathsf{S}, h) \models \varphi_i$ for $i \in \{1, 2\}$.
$(\mathsf{S}, h) \models \langle\!\langle A\rangle\!\rangle\psi$ iff for some joint strategy $\sigma_A \in \Sigma(\mathsf{S})$, $(\mathsf{S}, \pi, |h| - 1) \models \psi$ for all $\pi \in \mathsf{out}(h, \sigma_A)$.
$(\mathsf{S}, h) \models \mathbb{K}_a\varphi$ iff for every history $h' \in \mathsf{hist}(\mathsf{S})$, $h' \equiv_a h$ implies $(\mathsf{S}, h') \models \varphi$.
$(\mathsf{S}, h) \models \mathbb{C}_A\varphi$ iff for every history $h' \in \mathsf{hist}(\mathsf{S})$, $h' \equiv_A^{\mathbb{C}} h$ implies $(\mathsf{S}, h') \models \varphi$.
$(\mathsf{S}, h) \models \mathbb{D}_A\varphi$ iff for every history $h' \in \mathsf{hist}(\mathsf{S})$, $h' \equiv_A^{\mathbb{D}} h$ implies $(\mathsf{S}, h') \models \varphi$.
$(\mathsf{S}, \pi, m) \models \varphi$ iff $(\mathsf{S}, \pi_{\leq m}) \models \varphi$, for $\varphi$ a history formula.
$(\mathsf{S}, \pi, m) \models \neg\psi$ iff $(\mathsf{S}, \pi, m) \not\models \psi$.
$(\mathsf{S}, \pi, m) \models \psi_1 \wedge \psi_2$ iff $(\mathsf{S}, \pi, m) \models \psi_i$ for $i \in \{1, 2\}$
$(\mathsf{S}, \pi, m) \models \mathsf{X}\,\psi$ iff $(\mathsf{S}, \pi, m + 1) \models \psi$.
$(\mathsf{S}, \pi, m) \models \psi_1\,\mathsf{U}\,\psi_2$ iff for some $j \geq m$, $(\mathsf{S}, \pi, j) \models \psi_2$, and for all $k$ with $m \leq k < j$, we have $(\mathsf{S}, \pi, k) \models \psi_1$.

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

We isolate some important fragments.

1. The fragment $\mathsf{ATL}_{\mathbb{K},\mathbb{C},\mathbb{D}}$ consists of history formulas $\varphi$ defined by the grammer above, except with the following path formulas: $\psi ::= \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi$

2. The fragment $\mathsf{ATL}$ (resp. $\mathsf{ATL}^*$) consists of formulas of $\mathsf{ATL}_{\mathbb{K},\mathbb{C},\mathbb{D}}$ (resp. $\mathsf{ATL}^*_{\mathbb{K},\mathbb{C},\mathbb{D}}$) that do not mention epistemic operators.

3. The $\mathsf{CTL}$ operator $\mathsf{E}$ (resp. $\mathsf{A}$) is definable in $\mathsf{ATL}^*$ by $[[\emptyset]]$ (resp. $\langle\!\langle\emptyset\rangle\!\rangle$). In particular, $\mathsf{CTL}^*_{\mathbb{K},\mathbb{C},\mathbb{D}}$ is a syntactic fragment of $\mathsf{ATL}^*_{\mathbb{K},\mathbb{C},\mathbb{D}}$. The fragment of $\mathsf{CTL}^*_{\mathbb{K},\mathbb{C},\mathbb{D}}$ consisting of formulas of the form $\mathsf{A}\,\psi$, where $\psi$ is a path formula, is denoted $\mathsf{LTL}_{\mathbb{K},\mathbb{C},\mathbb{D}}$. Finally, $\mathsf{LTL}$ is the fragment of $\mathsf{LTL}_{\mathbb{K},\mathbb{C},\mathbb{D}}$ that does not mention epistemic operators.

Remark 1. *The definition of the semantics of $\langle\!\langle A\rangle\!\rangle\psi$ is the "objective" semantics of $\langle\!\langle A\rangle\!\rangle$, 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 $\mathcal{C}$ be a class of iCGS and $\mathcal{F}$ a sublanguage of $\mathsf{ATL}^*_{\mathbb{K},\mathbb{C},\mathbb{D}}$.* Model checking $\mathcal{C}$ against $\mathcal{F}$ specifications *is the following decision problem: given $\mathsf{S} \in \mathcal{C}$ and $\varphi \in \mathcal{F}$ as input, decide whether $\mathsf{S} \models \varphi$.*

Model checking is undecidable in general. Actually, it is undecidable even if $\mathcal{C}$ consists of all iCGS with $|Ag| = 3$ and $\mathcal{F}$ consists of the single formula $\langle\!\langle\{1, 2\}\rangle\!\rangle\,\mathsf{G}\,p$, see [6]. In Section 3 we prove that model checking PA-iCGS against $\mathsf{ATL}^*_{\mathbb{K},\mathbb{C},\mathbb{D}}$ 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 $\mathsf{ATL}^*_{\mathbb{K},\mathbb{C},\mathbb{D}}$.

**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 $Ag = \{F\} \cup \{1, 2, \cdots, n\}$. We give the father two actions, i.e., $Act_F = \{\exists \mathsf{m}, \neg\exists \mathsf{m}\}$. We give the $i$th child two actions, i.e., $Act_i = \{\mathsf{know\_muddy}, \neg\mathsf{know\_muddy}\}$. The set of states is $S = \{0,1\}^n \times (\mathsf{ACT} \cup \{\epsilon\})$. 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 ($\epsilon$ 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 $\sim_i = \{((v, J), (v', J')) : \bigwedge_{j \neq i} v_j = v'_j, J = J'\}$. Finally, the atoms are $AP = \{m_i : i \leq n\} \cup \{\alpha_a : \alpha \in Act_a, a \in Ag\}$. 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 $\mathsf{Kw}_a\phi$, read "agent $a$ knows whether $\phi$", by the formula $\mathbb{K}_a\phi \vee \mathbb{K}_a\neg\phi$. Consider the formula:

$$\langle\!\langle \{f, 1, \ldots, n\} \rangle\!\rangle [(\bigwedge \mathsf{G}(\mathsf{X}\, \alpha_a \rightarrow \widehat{\alpha_a})) \wedge \mathsf{F}(\bigwedge_{i \leq n} \mathsf{Kw}_i m_i)]$$

where the first conjunction is over $a \in Ag, \alpha \in Act_a$, and $\widehat{\alpha_a}$ is a formula representing the intended meaning of $\alpha_a$, e.g., if $\alpha_a = \mathsf{know\_muddy}$ then $\widehat{\alpha_a} = \mathsf{Kw}_a m_a$. 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:

$$\langle\!\langle \{f\} \rangle\!\rangle\, \mathsf{F} \bigwedge_{i \leq n} \mathsf{Kw}_i m_i$$

This formula is true. Indeed, a strategy for the father is to play action $\exists \mathsf{m}$ 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 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 $10 \times 10$-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 $win_i$ and $lose_i$ 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:

$$\langle\!\langle \{1, 2\} \rangle\!\rangle\, \mathsf{F} \langle\!\langle \{1, 3\} \rangle\!\rangle\, \mathsf{F}(lose_2 \vee \langle\!\langle \{1\} \rangle\!\rangle\, \mathsf{F}\, win_1)$$

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 $\mathsf{ATL}^*_{\mathbb{K},\mathbb{C},\mathbb{D}}$ specifications is decidable. This should be contrast with the fact that model checking arbitrary iCGS against $\mathsf{ATL}$ specifications is undecidable [6].

THEOREM 1. *Model checking PA-iCGS against* $\mathsf{ATL}^*_{\mathbb{K},\mathbb{C},\mathbb{D}}$ *specifications is* 2EXPTIME*-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* $\mathsf{S}$ *be an iCGS. Let* $\mu : S_0 \times \mathsf{ACT}^* \to \mathsf{hist}(\mathsf{S})$ *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* $u \in \mathsf{ACT}^*$*. That is,* $\mu(s_0, u)$ *is the history* $h$ *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* $\mathsf{S}$ *be a PA-iCGS. Since each* $\sim_a$ *is reflexive,* $\delta(s, \cdot) : \mathsf{ACT} \to S$ *is injective for every* $s \in S$*. Thus,* $\mu : S_0 \times \mathsf{ACT}^* \to \mathsf{hist}(\mathsf{S})$ *is a bijection. In particular, for every* $h \in \mathsf{hist}(\mathsf{S})$ *and* $s \in S_0$ *there exists a unique* $u \in \mathsf{ACT}^*$ *such that* $\mu(s, u) = h$*. This bijection allows us to encode histories of* $\mathsf{S}$ *by (unique) elements of* $S_0 \times \mathsf{ACT}^*$*.*

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* $\mathsf{S}$ *be a PA-iCGS. For all* $a \in Ag$*,* $u, u' \in \mathsf{ACT}^*$ *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 successfuly applied to a number of logics, including $\mathsf{CTL}^*$ [19] and $\mathsf{ATL}^*$ [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 *determinstic 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*, $\rho : S \times \Sigma \to S$

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 $s_0 s_1 \cdots s_{|u|}$ where $\rho(s_i, u_i) = s_{i+1}$ 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 DFW is *empty* if it accepts no word. A set of strings $X \subseteq \Sigma^*$ is called *regular* if there is a DFW $M$ 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 automata (*DPW*)* is a tuple $P = (\Sigma, S, s_0, \rho, c)$ where all components are as for DFW except that $c : S \to \mathbb{Z}$ is the *colouring* function. The run of $P$ on $\alpha \in \Sigma^\omega$ is the infinite sequence $s_0 s_1 \cdots$ such that $\rho(s_i, \alpha_i) = s_{i+1}$ for all $i$. The automaton *accepts* a word $\alpha$ iff the smallest color $k$ for which there are infinitely many $i$ with $c(s_i) = k$ is even (where $s_0 s_1 \cdots$ is the run of $M$ on $\alpha$).

We also make use of automata operating on trees. A *deterministic parity tree automata (*DPW*)* is a tuple $T = (\Sigma, D, S, s_0, \rho, c)$ where all components are as for a DPW except that $D$ is the finite set of *directions*, and $\rho : S \times \Sigma \to S^D$. The automaton operates on $\Sigma$-labelled $D$-ary branching *trees*, i.e., functions $f : D^* \to \Sigma$. 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 $g : D^* \to Q$ 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 $i$ with $g(\beta_i) = k$ is even.

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

PROPOSITION 1 ([35, 29]). *Every* LTL *formula $\psi$ over atoms $AP$ can be effectively converted into a* DPW *$P_\psi$ with input alphabet $2^{AP}$ that accepts a word $\alpha \in 2^{AP}$ iff $\alpha \models \psi$. Moreover, the* DPW *has double-exponentially many states and single-exponentially many colours.*

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

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

DEFINITION 6. *A* generalised iCGS *is a tuple*

$$\mathsf{S} = \langle Ag, AP, \{Act_a\}_{a \in Ag}, S, S_0, \delta, \{\sim_a\}_{a \in Ag}, \Lambda \rangle$$

*where all entries are as for iCGS, except that $\lambda : AP \to 2^S$ is replaced by a function $\Lambda : AP \to 2^{\mathsf{hist}(\mathsf{S})}$ such that $\Lambda(p) \subseteq \mathsf{hist}(\mathsf{S})$ is a regular set of histories, i.e., there exists a* DFW *over the alphabet $S$ accepting $h \in \mathsf{hist}(\mathsf{S})$ iff $h \in \Lambda(p)$.*

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

LEMMA 2. *Let $\mathsf{S}$ be an iCGS with labeling function $\lambda$. The generalised iCGS $\mathsf{S}'$ with labeling $\Lambda(p) = \{h \in \mathsf{hist}(\mathsf{S}) : last(h) \in \lambda(p)\}$ has the property that $\mathsf{S} \models \varphi$ iff $\mathsf{S}' \models \varphi$ (for all formulas $\varphi$ of $\mathsf{ATL}^*_{\mathbb{K},\mathbb{C},\mathbb{D}}$).*

PROOF. First, note that $\Lambda(p)$ is regular since a DFW 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 $\mathsf{S} \models \varphi$ iff $\mathsf{S}' \models \varphi$ holds by a straightforward induction on the structure of $\varphi$. □

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

**Inductive Statement**. Let $\mathsf{S}$ be a generalised PA-iCGS. For every history formula $\varphi$ and initial state $s \in S_0$ we will build a DFW $M^s_\varphi$ such that for every $u \in \mathsf{ACT}^*$,

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

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

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

**$\varphi$ is an atom**. Say $\varphi = p \in AP$ and let $s \in S_0$. The required DFW $M^s_\varphi$ should accept $u \in \mathsf{ACT}^*$ iff $\mu(s, u)$ is accepted by the DFW $\Lambda(p)$. To do this we define $M^s_\varphi$ to simulate $\mathsf{S}$ and the DFW $R = (S, Q, q_0, \rho, F)$ for $\Lambda(p)$ in parallel, i.e., by taking a product of $\mathsf{S}$ and $R$. Formally, define $M^s_\varphi = (\mathsf{ACT}, S \times Q, (\iota, q_0), \tau, F')$ where the transition function $\tau$ maps state $(s, q)$ and input $a \in \mathsf{ACT}$ to state $(\delta(s, a), \rho(q, s))$, and the final states $F'$ are of the form $(s, q)$ where $\rho(q, s) \in F$.

**$\varphi$ is a Boolean combination**. Let $s \in S_0$. The Boolean combinations follow from the effective closure of DFW under complementation and intersection. Indeed, $M^s_{\neg\varphi}$ is formed by complementing the final states of $M^s_\varphi$, and $M^s_{\varphi_1 \wedge \varphi_2}$ is the product of the $M^s_{\varphi_i}$s.

**$\varphi$ is of the form $\mathbb{K}_a \varphi'$**. Let $s \in S_0$. By induction, we have DFW $M^t_{\varphi'}$ for $t \in S_0$. The required DFW should accept a string $u$ iff, for every $t \in S_0$, if $\mu(s, u) \equiv_a \mu(t, u)$ then $M^t_{\varphi'}$ accepts $u$.

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

- $f_u(t) = last(\mu(t, u))$
- $t \in G_u$ iff for every prefix $v$ of $u$, $f_v(s) \sim_a f_v(t)$ (thus, initially $G_\epsilon = \{t \in S_0 : t \sim_a s\}$, and the moment $f_v(s) \not\sim_a f_v(t)$ we remove $t$ from $G_v$).

A state $\langle f, G, q\rangle$ is final if, for every $t \in G$ it is also the case that $q(t)$ is a final state of $M^t_\varphi$.

$\varphi$ **is of the form** $\mathbb{C}_A\varphi'$. This is identical to the $\mathbb{K}_a$ case except replace $\sim_a$ by $\sim^{\mathbb{C}}_A$ and replace $\equiv_a$ by $\equiv^{\mathbb{C}}_A$.

$\varphi$ **is of the form** $\mathbb{D}_A\varphi'$. This is identical to the $\mathbb{K}_a$ case except replace $\sim_a$ by $\sim^{\mathbb{D}}_A$ and replace $\equiv_a$ by $\equiv^{\mathbb{D}}_A$.

$\varphi$ **is of the form** $\langle\langle A\rangle\rangle\psi$. 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 DFW.

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

We briefly discuss the translation. Let $max(\psi)$ be the set of history subformulas of $\psi$ that are maximal, i.e., a history formula $\varphi \in max(\psi)$ iff it occurs in $\psi$ and that occurence is not a subformula of any other occurence of a history subformula of $\psi$. If $\psi$ is a path formula, define $lin(\psi)$ to be the LTL formula where for each $\varphi \in max(\psi)$ there is a fresh atom $\boxed{\varphi}$ such that every occurence of $\varphi$ in $\psi$ is replaced by $\boxed{\varphi}$. For example, consider $\psi = (p\,\mathsf{U}\langle\langle A\rangle\rangle\mathbb{K}_a p) \vee \mathsf{X}\neg\mathbb{C}_A p$. The history subformulas of $\psi$ are $\{p, \langle\langle A\rangle\rangle\mathbb{K}_a p, \mathbb{K}_a p, \neg\mathbb{C}_A p, \mathbb{C}_A p\}$. Then $max(\psi) = \{p, \langle\langle A\rangle\rangle\mathbb{K}_a p, \neg\mathbb{C}_A p\}$.[1] Thus $lin(\psi)$ is the LTL formula $\left(\boxed{p}\,\mathsf{U}\,\boxed{\langle\langle A\rangle\rangle\mathbb{K}_a p}\right) \vee \mathsf{X}\boxed{\neg\mathbb{C}_A p}$ over the atoms $\boxed{\varphi}$ for $\varphi \in max(\psi)$. Since, by induction, we have built DFW for each boxed atom, for the remainder of the proof we assume that $\psi$ is an LTL formula.

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

*Construction of* DPW $E_{\psi,s}$. To build $E_{\psi,s}$, begin by writing $AP(\psi)$ for the finite set of atoms appearing in $\psi$. First, for each $p \in AP(\psi)$, let $D^p$ be a DFW for the regular set $\{u \in \mathsf{ACT}^* : \mu(s,u) \in \Lambda(p)\}$. Second, by Proposition 1, every LTL formula $\psi$ can be converted into a DPW $D_\psi$ over alphabet $2^{AP}$ that accepts all word models of that formula. Let $Q_\psi$ be the states and $\Delta_\psi : Q_\psi \times 2^{AP(\psi)} \to Q_\psi$ the transition of the DPW. Now, the DPW $E_{\psi,s}$ simulates $D_\psi$ and each DFW $D^p$. The automaton does this by storing and updating a state $q_u \in Q_\psi$ and a function $f_u$ such that $f_u(p)$ is the state of $D^p$ (i.e., $f_u : AP(\psi) \to Q$ where $Q$ is the union of states of the $D^p$s). Transitions of $E_{\psi,s}$ are as follows: from state $\langle q_u, f_u\rangle$ and input $d \in \mathsf{ACT}$ the next state $\langle q_{ud}, f_{ud}\rangle$ satisfies that $q_{ud} = \Delta_\psi(q_i, Z)$ 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 the transition function of $D^p$ to the state $f_u(p)$ and input $d$. Define the colour of $\langle q, f\rangle$ to be the same as the colour in $D_\psi$ of the state $q$, and $\langle q, f\rangle$ is an initial state if $q$ is an initial state in $D_\psi$. The following is straightforward to prove:

LEMMA 3. *The* DPW $E_{\psi,s}$ *accepts* $\alpha \in \mathsf{ACT}^\omega$ *if and only if* $(\mathsf{S}, \mu(s,\alpha)) \models \psi$.

*Construction of* DPT $F_{\psi,s,q}$. To solve the synthesis problem we will encode strategies as trees and use tree-automata. Encode a strategy $\sigma$ of agent $a$ by the ACT-branching tree

$$T_\sigma : \mathsf{ACT}^* \to (Act_a)^{S_0}$$

where, for $u \in \mathsf{ACT}^*$, $T_\sigma(u)(t) = \sigma(\mu(t,u))$. By Lemma 1, $\sigma$ is uniform iff $T_\sigma$ satisfies the property

$$\mu(t,u) \equiv_a \mu(t',u) \Rightarrow T_\sigma(u)(t) = T_\sigma(u)(t') \qquad (1)$$

LEMMA 4. *The set of encodings $T_\sigma$ of uniform strategies $\sigma$ of agent $a$ is recognised by a* DPT.

PROOF. To do this, it is sufficient to build a DPT that accepts a tree $T$ iff $T$ has property (1). Informally, for every pair of different states $t, t' \in S_0$, the DPT keeps a bit that is initialised to 1 (signifying that it must verify that $T(u)(t) = T(u)(t')$), and as soon as it finds that $\mu(t,u) \not\equiv_a \mu(t',u)$ it sets the bit to 0, which signals that it no longer has to ensure $T_\sigma(u)(t) = T_\sigma(u)(t')$. □

Encode a set of uniform strategies $\sigma_A$ (for $A \subseteq Ag$) as the convolution $T_{\sigma_A}$ of their individual encodings, i.e.,

$$T_{\sigma_A} : \mathsf{ACT}^* \to \prod_{a\in A} (Act_a)^{S_0}$$

where $T_{\sigma_A}(u)(a) = T_{\sigma_a}(u)$. Running the automata for $\sigma_a$ in parallel yields:

LEMMA 5. *For every $A \subseteq Ag$, the set of encodings of joint uniform strategies $\sigma_A$ is recognised by a* DPT.

Finally, in order to solve the synthesis problem for state $q$, we define a DPT $F_{\psi,s,q}$ that accepts $T_{\sigma_A}$ iff for every $\alpha \in \mathsf{ACT}^\omega$, if $\mu(s,\alpha)$ is consistent with $\sigma_A$ then $\alpha$ is accepted by $E_{\psi,s}$ starting from $q$.

The DPT $F_{\psi,s,q}$ works as follows: it reads $T_{\sigma_A}$ as input and, on every branch $\alpha \in \mathsf{ACT}^\omega$ of the tree, simulates $E_{\psi,s}$ starting from $q$, and accepts that branch if both $E_{\psi,s}$ is accepting and $\mu(s,\alpha)$ is consistent with $\sigma_A$. This can be done by a tree-automaton because a) $E_{\psi,s}$ is deterministic and thus it can be run on all paths of the tree (i.e., this step would not be possible if $E_{\psi,s}$ were a non-deterministic automaton); and b) checking if $\mu(s,\alpha)$ is consistent with $\sigma_A$ is simply a matter of checking that $\alpha_0(a) = \sigma_A(a)(s)$ and $\alpha_{i+1}(a) = \sigma_A(a)(\mu(s, \alpha_{\leq i}))$, for every $i \in \mathbb{N}$ and $a \in A$.

*Final states of* DFW $M^s_{\langle\langle A\rangle\rangle\psi}$. Finally, define the state $q$ of $M^s_{\langle\langle A\rangle\rangle\psi}$ to be a final state iff the DPT $F_{\psi,s,q}$ is non-empty (a decidable condition).

This completes the construction of $M^s_{\langle\langle A\rangle\rangle\varphi}$, and the proof of decidability.

## 3.1 Complexity

In this section we analyse the complexity of our decision procedure for a fixed number of agents. We then establish the lower bound by a reduction from a problem known to be 2EXPTIME-hard.

First, we calculate $||\varphi||$, the number of states of $M^s_\varphi$, for each case:

[1]Note that although $p$ has a non-maximal occurence in $\psi$, it is included in $max(\psi)$ since it has at least one occurence 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: $||\varphi \wedge \varphi'|| = ||\varphi|| \times ||\varphi'||$.
4. Epistemic: $||\mathbb{K}_a\varphi|| = (2 \times |S| \times ||\varphi||)^{|S|}$, and the same for $||\mathbb{C}_A\varphi||$ and $||\mathbb{D}_A\varphi||$.
5. Strategic: $||\langle\langle A\rangle\rangle\psi|| = 2^{2^{O(|lin(\psi)|)}} \times ||\tilde{\psi}||^{|AP(lin(\psi))|} \times O(2^{|S|^2})$ where $lin(\psi)$ is the linearisation of $\psi$, $\tilde{\psi}$ is the largest history subformula of $\psi$, and $AP(\cdot)$ is the set of atomic predicates occuring 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 DPW $D_\psi$ has double-exponentially many states and single-exponentially many colours (in the size of $lin(\psi)$) [35, 29]. The DPW $E_{\psi,s}$ has $2^{2^{O(|lin(\psi)|)}} \times ||\tilde{\psi}||^{AP(\psi)}$ many states. The DPT $F_{\psi,s,q}$ has $2^{2^{O(|lin(\psi)|)}} \times ||\tilde{\psi}||^{AP(\psi)} \times O(2^{|S|^2})$ many states and $2^{O(|lin(\psi)|)}$ many colours. This gives the stated value of $||\langle\langle A\rangle\rangle\psi||$.

Second, we calculate the time for constructing the DFW $M^s_\varphi$. In the first four cases this cost is polynomial in the size of the DFW, i.e., $||\varphi||$. For the strategy case we incur a cost to calculate the final states, i.e., solving the emptiness of the DPT $F_{\psi,s,q}$. The cost of solving the emptiness of a DPT with $n$ states and $m$ colours is at most $n^{O(m)}$ [16]. Thus, the time for constructing $M^s_\varphi$ is at most $n^{O(m)}$ where $n = 2^{2^{O(|lin(\psi)|)}} \times ||\tilde{\psi}||^{AP(lin(\psi))} \times O(2^{|S|^2})$ and $m = 2^{O(|lin(\psi)|)}$.

Finally, let $z = |S| + |\varphi|$. The time for constructing $M^s_\varphi$ of each step of the procedure can be bounded above by $2^{2^{O(z)}} \times |\Lambda|^{2^{O(z)}}$ where $|\Lambda|$ 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 $z$ such steps, the time for constructing, and thus the size, of the resulting DFW is $2^{2^{O(z^2)}}$. Testing if $\epsilon$ is accepted by this automaton has no additional cost. Thus, our algorithm runs in 2EXPTIME.

**Lower-Bound**. To prove the lower bound in Theorem 1 we reduce from a known 2EXPTIME-hard problem, i.e., model checking a CGS with $Ag = \{a, b\}$ against a formula of the form $\langle\langle\{a\}\rangle\rangle\psi$ for an LTL formula $\psi$ [27, 28]. One can translate a two-player CGS $\mathsf{S}$ into a polynomially larger CGS $\mathsf{S}'$ with public actions such that $\mathsf{S} \models \varphi$ iff $\mathsf{S}' \models \varphi$. Indeed, the agents, actions, and atoms are the same, $S' = S \times (\mathsf{ACT} \cup \{\epsilon\})$, $S'_0 = S_0 \times \{\epsilon\}$, $\delta'((s,d),d') = (\delta(s,d'),d')$, and $\lambda'(p) = \{(s,d) : s \in \lambda(p)\}$ (note that because $\mathsf{S}$ has two-players, $|\mathsf{ACT}| = |Act_1| \times |Act_2|$, and thus $|\mathsf{S}'|$ is polynomial in the size of $\mathsf{S}$).

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

The decidability proof of Theorem 1 is for objective semantics. To deal with subjective semantics proceed as follows. For $u \in \mathsf{ACT}^*$ let $T^s_u \subseteq S_0$ be the set of $t$ such that there exists $a \in A$ with $t \equiv_a s$. The DFW $M^s_{\langle\langle A\rangle\rangle\psi}$ simulates $E_{\psi,t}$ for all $t \in T^s_\epsilon$. After reading $u$, each automaton $E_{\psi,t}$ for $t \in T^s_u$ is 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 $\alpha \in \mathsf{ACT}^*$ and $t \in T^s_u$, if $\mu(t, u\cdot\alpha) \in \mathsf{out}(\mathsf{S}, \mu(t,u), \sigma_A)$ then $\alpha$ 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 $t \in T^s_u$ (as we did above for $s$). Then we check if the intersection of the automata $F_{\psi,t,q_t}$ (for $t \in T^s_u$) 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 $\mathsf{ATL}^*_\mathbb{K}$ 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* $\mathsf{ATL}^*_\mathbb{K}$. We show how to reduce the realisability problem for LK to the model checking problem for $\mathsf{ATL}^*_\mathbb{K}$. 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:

$$\psi \quad ::= \quad p \mid \neg\psi \mid \psi \wedge \psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi \mid \mathbb{K}_a\psi$$

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

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

$(\mathsf{S}, \pi, m) \models_{\mathsf{LK}} p$ iff $\pi_m \in \lambda(p)$
$(\mathsf{S}, \pi, m) \models_{\mathsf{LK}} \neg\psi$ iff $(\mathsf{S}, \pi, m) \not\models_{\mathsf{LK}} \psi$
$(\mathsf{S}, \pi, m) \models_{\mathsf{LK}} \psi_1 \wedge \psi_2$ iff $(\mathsf{S}, \pi, m) \models_{\mathsf{LK}} \psi_i$ for $i \in \{1, 2\}$
$(\mathsf{S}, \pi, m) \models_{\mathsf{LK}} \mathsf{X}\,\psi$ iff $(\mathsf{S}, \pi, m+1) \models_{\mathsf{LK}} \psi$
$(\mathsf{S}, \pi, m) \models_{\mathsf{LK}} \psi_1\,\mathsf{U}\,\psi_2$ iff for some $j \geq m$, $(\mathsf{S}, \pi, j) \models_{\mathsf{LK}} \psi_2$, and for all $k$ with $m \leq k < j$, $(\mathsf{S}, \pi, k) \models_{\mathsf{LK}} \psi_1$

The epistemic operator $\mathbb{K}_a$ is follows:

$(\mathsf{S}, \pi, m) \models_{\mathsf{LK}} \mathbb{K}_a\psi$ iff $(\mathsf{S}, \pi', m') \models_{\mathsf{LK}} \psi$ for all $\pi' \in \mathsf{comp}(\mathsf{S})$ such that $\pi_{\leq m} \equiv_a \pi'_{\leq m'}$ (in particular, $m = m'$).

An LK-formula $\psi$ is *realisable* if there exists a uniform strategy $\sigma_A$ such that for all $s_0 \in S_0$ and all $\pi \in \mathsf{out}(\mathsf{S}, s_0)$, we have that $(\mathsf{S}, \pi, 0) \models \psi$.

We now present the reduction. Let $\psi$ be a formula of LK. Define $\hat{\psi}$, a path formula of $\mathsf{CTL}^*_\mathbb{K}$, by recursively replacing $\mathbb{K}_a\psi$ by $\mathbb{K}_a\,\mathsf{A}\,\hat{\psi}$. We claim that for all iCGS $\mathsf{S}$, $\psi$ is realisable in $\mathsf{S}$ iff $\mathsf{S} \models \langle\langle Ag\rangle\rangle\hat{\psi}$. To see this it is sufficient to establish the following inductive hypothesis: $(\mathsf{S}, \pi, m) \models_{\mathsf{LK}} \psi$ iff $(\mathsf{S}, \pi, m) \models \hat{\psi}$. To prove the inductive hypothesis use that the following are equivalent to $(\mathsf{S}, \pi, m) \models_{\mathsf{LK}} \mathbb{K}_a\psi$:

1. $(\mathsf{S}, \pi', m) \models_{\mathsf{LK}} \psi$ for all $\pi' \in \mathsf{comp}(\mathsf{S})$ such that $\pi'_{\leq m} \equiv_a \pi_{\leq m}$ (by the definition of $\models_{\mathsf{LK}}$);

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

3. $(\mathsf{S}, \pi, m) \models \mathbb{K}_a \mathsf{A} \hat{\psi}$ (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 $Ag = \{e, 1, 2, \cdots, n\}$ whose state set is of the form $S = L_e \times \prod_{i \leq n} L_i$ (for some finite sets $L_a$ for $a \in Ag$), and whose transition maps state $(l_e, l_1, \cdots, l_n)$ and joint-action $J \in \mathsf{ACT}$ to the state $(\tau_e(l_e, J), \tau_1(l_1, J(1)), \cdots, \tau_n(l_n, J(n)))$ where $\tau_e : L_e \times \mathsf{ACT} \to L_e$ and $\tau_i : L_i \times Act_i \to L_i$ 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 $(l_e, l_1, \cdots, l_n) \sim_i (l'_e, l'_1, \cdots, l'_n)$ iff $l_i = l'_i$ and $F(l_e) = F(l'_e)$ where $F : L_e \to O$ is a function mapping environment states to some fixed set $O$ of observations (note that $F$ and $O$ are independent of $i$, 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 $L_e = \mathsf{ACT} = O$ 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 $\mathsf{ATL}^*_{\mathbb{K},\mathbb{C},\mathbb{D}}$ of the alternating-time temporal logic $\mathsf{ATL}^*$, 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 $\mathsf{ATL}$ 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 $\mathsf{ATL}^*$ 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 $\langle\langle A \rangle\rangle \psi$ is true then there are finite-memory strategies witnessing this fact, and if a formula $\mathbb{K}_a \varphi$ is true then there is a finite-state machine that accepts exactly the histories making $\mathbb{K}_a \varphi$ 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 $\mathsf{ATL}^*_{\mathbb{K},\mathbb{C},\mathbb{D}}$ 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

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

[2] D. Berwanger, A. B. Mathew, and M. van den Bogaard. Hierarchical Information Patterns and Distributed Strategy Synthesis. In *ATVA'15*, LNCS 9364, pages 378–393. Springer, 2015.

[3] N. Bulling and W. Jamroga. Comparing variants of strategic ability: how uncertainty and memory influence general properties of games. *Journal of Autonomous agents and multi-agent systems*, 28(3):474–518, 2014.

[4] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, 1999.

[5] C. Dima, C. Enea, and D. Guelev. Model-checking an alternating-time temporal logic with knowledge, imperfect information, perfect recall and communicating coalitions. In *GandALF 2010*, volume 25 of *EPTCS*, pages 103–117, 2010.

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

[7] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[8] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[9] B. Finkbeiner and S. Schewe. Uniform Distributed Synthesis. In *LICS'05*, pages 321–330. IEEE Computer Society, 2005.

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

[11] D. P. Guelev, C. Dima, and C. Enea. An alternating-time temporal logic with knowledge, perfect recall and past: axiomatisation and model-checking. *Journal of Applied Non-Classical Logics*, 21(1):93–131, 2011.

[12] J. Gutierrez, P. Harrenstein, and M. Wooldridge. Reasoning about equilibria in game-like concurrent systems. In *KR'14*. AAAI, 2014.

[13] J. Gutierrez, P. Harrenstein, and M. Wooldridge. Expresiveness and complexity results for strategic reasoning. In *CONCUR'15*, volume 42 of *LIPIcs*, 2015.

[14] W. Hoek and M. Wooldridge. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75(1):125–157, 2003.

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

[16] M. Jurdzinski. Small Progress Measures for Solving Parity Games. In *STACS'00*, LNCS 1770, pages 290–301. Springer, 2000.

[17] P. Kazmierczak, T. Ågotnes, and W. Jamroga. Multi-agency is coordination and (limited) communication. In *PRIMA'14*, LNCS 8861, pages 91–106. Springer, 2014.

[18] O. Kupferman and M. Vardi. Synthesizing Distributed Systems. In *LICS'01*, pages 389–398. IEEE Computer Society, 2001.

[19] O. Kupferman, M. Vardi, and P. Wolper. An Automata Theoretic Approach to Branching-Time Model Checking. *Journal of the ACM*, 47(2):312–360, 2000.

[20] O. Kupferman, M. Vardi, and P. Wolper. Module Checking. *Information and Computation*, 164(2):322–344, 2001.

[21] F. Laroussinie, N. Markey, and A. Sangnier. ATLsc with partial observation. In *GandALF 2015*, volume 193 of *EPTCS*, pages 43–57, 2015.

[22] A. Lomuscio, H. Qu, and F. Raimondi. MCMAS: A model checker for the verification of multi-agent systems. *Software Tools for Technology Transfer*, 19(1):9–30, 2017.

[23] A. R. Lomuscio, R. van der Meyden, and M. Ryan. Knowledge in multiagent systems: Initial configurations and broadcast. *ACM Trans. Comput. Logic*, 1(2):247–284, Oct. 2000.

[24] A. Lopes, F. Laroussinie, and N. Markey. ATL with Strategy Contexts: Expressiveness and Model Checking. In *FSTTCS'10*, LIPIcs 8, pages 120–132. Leibniz-Zentrum fuer Informatik, 2010.

[25] F. Mogavero, A. Murano, G. Perelli, and M. Vardi. Reasoning About Strategies: On the Model-Checking Problem. *TOCL*, 15(4):34:1–42, 2014.

[26] H. P. Nii. Blackboard systems, part one: The blackboard model of problem solving and the evolution of blackboard architectures. *AI Magazine*, 7(2):38–53, 1986.

[27] A. Pnueli and R. Rosner. Distributed Reactive Systems Are Hard to Synthesize. In *FOCS'90*, pages 746–757. IEEE Computer Society, 1990.

[28] R. Rosner. *Modular Synthesis of Reactive Systems*. PhD thesis, Weizmann Institute of Science, Rehovot, Israel, 1992.

[29] S. Safra. *Complexity of Automata on Infinite Objects*. PhD thesis, Weizmann Institute of Science, Rehovot, Israel, 1989.

[30] S. Schewe and B. Finkbeiner. Distributed synthesis for alternating-time logics. In K. S. Namjoshi, T. Yoneda, T. Higashino, and Y. Okamura, editors, *ATVA'07*, pages 268–283. Springer, 2007.

[31] P. Schobbens. Alternating-Time Logic with Imperfect Recall. *ENTCS*, 85(2):82–93, 2004.

[32] R. van der Meyden and H. Shilov. Model checking knowledge and time in systems with perfect recall. In *FST&TCS'99*, LNCS 1738, pages 432–445. Springer, 1999.

[33] R. van der Meyden and T. Wilke. Synthesis of distributed systems from knowledge-based specifications. In *CONCUR'05*, LNCS 3653, pages 562–576. Springer, 2005.

[34] H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*, volume 337 of *Synthese Library*. Springer, 2007.

[35] M. Y. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, 1994.

[36] S. Vester. Alternating-time temporal logic with finite-memory strategies. In *GandALF'13*, volume 119 of *EPTCS*, pages 194–207, 2013.