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

Francesco Belardinelli

IBISC, Université d'Evry, France

`belardinelli@ibisc.fr`

Aniello Murano

DIETI, Università degli Studi di Napoli

`murano@na.infn.it`

Alessio Lomuscio

Department of Computing, Imperial College London

`a.lomuscio@imperial.ac.uk`

Sasha Rubin

DIETI, Università degli Studi di Napoli

`rubin@unina.it`

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. This paper is based on a previous version [2] accepted for publication at AAMAS2017.

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 [12]. Reasoning about the knowledge and the strategic ability of agents in these systems remains of particular importance. Traditionally, epistemic logic [12] 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 [5].

However, verifying synchronous, perfect recall MAS under incomplete information against specifications in ATL is undecidable [1, 7] (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, LTL_K, the extension of LTL with individual-knowledge operators, as in [30]), restricting the semantics (e.g., by requiring strategy quantifiers to vary over memoryless-strategies [38]), 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 [7], we prove that model-checking the full logic ATL*K on PA-iCGS is decidable, specifically 2EXPTIME-complete. Thus, the joint complexity of model-checking is the same as that of 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 [31], blackboard systems in which a public database is read and written by agents [33], auctions and auction-based mechanisms [10].

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, 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. We conclude in Section 4.

Related Work. In order to reason formally about multi-agent systems, temporal logics such as LTL, CTL, CTL* have been extended with strategy quantifiers [1] and epistemic modalities [18]. 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, 18, 14, 38, 19, 6, 15, 32]. 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 [7], as is the model-checking problem for the extension LTL$_{K,C}$ of LTL with epistemic operators, including common knowledge [30]. 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$_K$ with only individual knowledge is non-elementary complete [30], model-checking ATL with only "communicating coalitions" (i.e., coalitions use their distributed knowledge instead of their individual knowledge) is decidable and non-elementary [6, 15]; and model-checking ATL in which all coalitions operate with a single indistinguishability relation reduces ATL to its singleton-coalition fragment [21]. 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 [25]; that the information sets form a hierarchy [34], or that, over time, they infinitely often form a hierarchy [3]. A notable example of (ii) are characterisations of the architectures for which distributed synthesis is decidable [13, 37], thus generalising earlier results on linear architectures [34, 22].

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. In particular, [31] studies the synthesis problem for MAS in broadcast environments and specifications given by linear-temporal logic with individual-knowledge operators. The main differences between this work and ours is as follows: our logic can express the synthesis problem as a model checking problem (use formulas of the form $\langle\langle A \rangle\rangle \psi$ where ψ is a path-formula that does not contain strategic quantifiers), as well as alternating strategic quantifiers mentioning overlapping coalitions; PA-iCGS can be viewed as a special case of broadcasting systems (in broadcasting systems each agent has the same observation of the environment, which can be take to be the last joint-action); the interpretation of the epistemic operators in their work are restricted to histories consistent with the strategies being synthesised, while in our work histories need only start in an initial state.

Actions that constitute public announcements have been studied in depth (Dynamic Epistemic Logic, Public Announcement Logic, epistemic protocols [8]). 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.

This paper is based on a previous version [2] presented at AAMAS2017.

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 ATL*K, an extension of Alternating Time Temporal Logic (ATL*) which includes strategic operators ($\langle\langle A \rangle\rangle$ for $A \subseteq$ Ag) as well as epistemic operators for individual-knowledge (K_a for $a \in$ Ag), common-knowledge (C_A for $A \subseteq$ Ag), and distributed-knowledge (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 ith element of u, i.e., $u = u_0 u_1 \cdots$. The empty sequence is denoted ε. The length of a finite sequence $u \in X^*$ is denoted $|u|$, its last (resp. first) element is denoted $last(u)$ (resp. $first(u)$). Further, we assume that $last(\varepsilon) = first(\varepsilon) = \varepsilon$. 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 ith co-ordinate of v by $v(i)$. In particular, for $F \in \prod_i (X_i)^Y$ we may write $F(i) \in (X_i)^Y$ 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 [4, 7].

Definition 1 (iCGS). *An* imperfect information concurrent game structure (iCGS) *is a tuple* $S = \langle Ag, AP, \{Act_a\}_{a \in Ag}, S, S_0, \delta, \{\sim_a\}_{a \in Ag}, \lambda \rangle$, *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;

- *$\delta : S \times$ ACT $\to S$ is the* transition function, *where the set* 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 $J, J' \in$ 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 $T \times (\mathsf{ACT} \cup \{\varepsilon\})$, where ε 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 [12].

In the remainder of this section we define what it means for an agent to have *synchronous perfect-recall* [12].

Synchronous perfect-recall under imperfect information. A *path* in 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 S is written $\mathsf{comp}(\mathsf{S})$, and the set of computations in 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 $A \subseteq \mathsf{Ag}$, let $\sim_A^C = (\cup_{a \in A} \sim_a)^*$, where $*$ denotes the reflexive transitive closure (wrt. composition of relations), and its extension to histories is denoted $\equiv_A^C$. For $A \subseteq \mathsf{Ag}$, let $\sim_A^D = \cap_{a \in A} \sim_a$, and its extension to histories is denoted $\equiv_A^D$.

Strategies. A *deterministic memoryfull strategy*, or simply a *strategy*, for agent a is a function $\sigma_a : \mathsf{hist}(\mathsf{S}) \to \mathsf{Act}_a$. A strategy σ_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 $A \subseteq \mathsf{Ag}$, let $\sigma_A : A \to \Sigma(\mathsf{S})$ denote a function that associates a uniform strategy σ_a with each agent $a \in A$. We write $\sigma_A(a) = \sigma_a$, and call σ_A a *joint strategy*. For $h \in \mathsf{hist}(\mathsf{S})$ write $\mathsf{out}(\mathsf{S}, h, \sigma_A)$, called the *outcomes of σ_A from h*, for the set of computations $\pi \in \mathsf{cmp}(\mathsf{S}, h)$ such that π is consistent with σ_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 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 σ_A is the empty function and (ii) above places no additional restrictions on the computations).

2.2 The Logic $\mathsf{ATL}^*\mathsf{K}$

In this section we define the logic $\mathsf{ATL}^*\mathsf{K}$. Its syntax has been called ATEL* (cf. [18]), and we interpret it on iCGS with history-based semantics and imperfect information.

Syntax. Fix a finite set *AP* of *atomic propositions (atoms)* and a finite set Ag of *agents*. The *history (φ) and path (ψ) formulas over AP and Ag* are built using the following grammar:

$$\varphi \quad ::= \quad p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_a\varphi \mid C_A\varphi \mid D_A\varphi \mid \langle\!\langle A \rangle\!\rangle \psi$$
$$\psi \quad ::= \quad \varphi \mid \neg\psi \mid \psi \wedge \psi \mid \mathsf{X}\psi \mid \psi \mathsf{U} \psi$$

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

The class of $\mathsf{ATL}^*\mathsf{K}$ *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 $\langle\!\langle A \rangle\!\rangle$ ("the agents in A can

enforce ψ"), 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"). Intuitively, $C_A\phi$ means that every agent $a \in A$ knows ϕ, and a knows that everybody in A knows ϕ, and so on. On the other hand, $D_A\phi$ means that if all agents in A pool their knowledge, then they know ϕ. We introduce operators $[[A]]$, G, and F as standard.

Semantics. Fix an iCGS S. We simultaneously define, by induction on the formulas, $(\mathsf{S},h) \models \varphi$ where $h \in \mathsf{hist}(\mathsf{S})$ and φ is a history formula, and $(\mathsf{S},\pi,m) \models \psi$ where $\pi \in \mathsf{cmp}(\mathsf{S})$, $m \geq 0$, and ψ 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 $\pi \in \mathsf{out}(h,\sigma_A)$.
$(\mathsf{S},h) \models K_a\varphi$	iff	for every history $h' \in \mathsf{hist}(\mathsf{S})$, $h' \equiv_a h$ implies $(\mathsf{S},h') \models \varphi$.
$(\mathsf{S},h) \models C_A\varphi$	iff	for every history $h' \in \mathsf{hist}(\mathsf{S})$, $h' \equiv_A^C h$ implies $(\mathsf{S},h') \models \varphi$.
$(\mathsf{S},h) \models D_A\varphi$	iff	for every history $h' \in \mathsf{hist}(\mathsf{S})$, $h' \equiv_A^D h$ implies $(\mathsf{S},h') \models \varphi$.
$(\mathsf{S},\pi,m) \models \varphi$	iff	$(\mathsf{S},\pi_{\leq m}) \models \varphi$, for φ 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 $m \leq k < j$, $(\mathsf{S},\pi,k) \models \psi_1$.

For a history formula φ, 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 $\mathsf{ATL}_{K,C,D}$ consists of history formulas φ defined by the grammar above, except with the following path formulas: $\psi ::= \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi$

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

3. The CTL operator E (resp. A) is definable in ATL^* by $[[\emptyset]]$ (resp. $\langle\langle\emptyset\rangle\rangle$). In particular, $\mathsf{CTL}^*_{K,C,D}$ is a syntactic fragment of $\mathsf{ATL}^*\mathsf{K}$. The fragment of $\mathsf{CTL}^*_{K,C,D}$ consisting of formulas of the form $\mathsf{A}\psi$, where ψ is a path formula, is denoted LTLK. Finally, LTL is the fragment of LTLK 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 [38]. To obtain the subjective interpretation of $\mathsf{ATL}^*\mathsf{K}$ replace "for all $\pi \in \mathsf{out}(h,\sigma_A)$" by "for all $\pi \in \bigcup_{a \in A, h' \equiv_a h} \mathsf{out}(h',\sigma_A)$" 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 K_a is definable in terms of $\langle\langle A \rangle\rangle$, i.e., $\langle\langle a \rangle\rangle \varphi \mathsf{U} \varphi$.*

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

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

Model checking is undecidable in general. Actually, it is undecidable even if $\mathscr{C}$ consists of all iCGS with $|\mathsf{Ag}| = 3$ and $\mathscr{F}$ consists of the single formula $\langle\langle\{1,2\}\rangle\rangle \mathsf{G} p$, see [7]. In Section 3 we prove that model checking PA-iCGS against $\mathsf{ATL}^*\mathsf{K}$ specifications is decidable.

Discussion. We emphasize that PA-iCGS are not only of theoretical interest, but they are indeed a family of systems relevant from the viewpoint of modelisation and verification of real-life scenarios.

Firstly, notice that several card and boards games can be characterised as games of imperfect information, in which only a bounded amount of private communication/actions (possibly none) is performed. Such examples of games include poker, bridge, battleship, in which the imperfect information pertains to the inital deal of cards/pawns, while all moves are visible to any player.

Further, several epistemic puzzles fall within the category of PA-iCGS, such as muddy children, Russian cards, consecutive numbers, sum and products [9]. We refer to [2] for a formalisation of the muddy children puzzle in the framework of *iCGS with only public actions*, which can be recast as a PA-iCGS.

Moreover, we observe that broadcast/public actions are amongst the most well-studied communication patterns in distributed computing, because of their relevance and ubiquity. For instance, broadcast is the default communication mode for Twitter. In this respect, PA-iCGS are closely related to (and able to capture) the broadcasting systems in [26], as well as planning via public actions [11].

3   Decidability of PA-iCGS

In this section we state that model checking PA-iCGS against ATL*K specifications is decidable. This should be contrasted with the fact that model checking arbitrary iCGS against ATL specifications is undecidable [7].

Theorem 5. *Model checking PA-iCGS against* ATL*K *specifications is* 2EXPTIME-*complete.*

The rest of this section contains the proof of this theorem. We use an automata-based marking algorithm. Such algorithms have been successfuly applied to a number of logics, including CTL* [23] and ATL* [1] in the perfect information setting. Before giving the proof, we introduce an encoding μ of histories.

Definition 6. *Let* S *be an iCGS. Let* $\mu : S_0 \times$ ACT$^* \to$ 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* $u \in$ 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* S *be a PA-iCGS. Since each* $\sim_a$ *is reflexive,* $\delta(s, \cdot) :$ ACT $\to S$ *is injective for every* $s \in S$. *Thus,* $\mu : S_0 \times$ ACT$^* \to$ hist(S) *is a bijection. In particular, for every* $h \in$ hist(S) *and* $s \in S_0$ *there exists a unique* $u \in$ ACT* *such that* $\mu(s, u) = h$. *This bijection allows us to encode histories of* S *by (unique) elements of* $S_0 \times$ 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, $u, u' \in$ 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 $j < |u|$. $\square$

We are ready to proceed with the proof.

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 [24].

A *deterministic finite-word automaton (*DFW*)* is a tuple $M = (\Sigma, S, s_0, \rho, F)$ where Σ 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 α 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 α).

We also make use of automata operating on trees. A *deterministic parity tree automata (*DPT*)* 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 Σ-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 $g(\varepsilon) = s_0$ and $g(ud) = \rho(g(u), t(u))$. The automaton *accepts* the tree f iff for every $\beta \in D^\omega$, 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 ([39, 36]). *Every* LTL *formula* ψ *over atoms AP can be effectively converted into a* DPW P_ψ *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 φ 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_φ^s 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 $\varphi = K_a \varphi'$ case is done by simulating the DFW $M_{\varphi'}^t$ 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 ψ is an LTL formula, and then we build a DPW for the formula ψ; 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 7. *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 ATL*K: $(\mathsf{S}, h) \models p$ iff $h \in \Lambda(p)$. It is immediate that generalised iCGS are indeed more general than iCGS:

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

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 $S \models \varphi$ iff $S' \models \varphi$ holds by a straightforward induction on the structure of φ. □

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 S as a generalised PA-iCGS.

INDUCTIVE STATEMENT. Let S be a generalised PA-iCGS. For every history formula φ and initial state $s \in S_0$ we will build a DFW M_φ^s such that for every $u \in \mathsf{ACT}^*$, M_φ^s accepts u iff $(S, \mu(s,u)) \models \varphi$.

From this it is easy to get the decidability stated in the theorem: simply check that ε is accepted by every M_φ^s with $s \in S_0$. We build the DFW M_φ^s, simultaneously for all $s \in S_0$, by induction on φ.

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

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

φ **is of the form** $K_a \varphi'$. Let $s \in S_0$. By induction, we have DFW $M_{\varphi'}^t$ 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 $M_{\varphi'}^t$ accepts u.

To do this, the DFW 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 $q : S_0 \to Q$ 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 $J \in \mathsf{ACT}$ 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 $u \in \mathsf{ACT}^*$ 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 $G_\varepsilon = \{t \in S_0 : t \sim_a s\}$, 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_φ^t.

φ **is of the form** $C_A \varphi'$. This is identical to the K_a case except replace $\sim_a$ by $\sim_A^C$ and replace $\equiv_a$ by $\equiv_A^C$.

φ **is of the form** $D_A \varphi'$. This is identical to the K_a case except replace $\sim_a$ by $\sim_A^D$ and replace $\equiv_a$ by $\equiv_A^D$.

φ **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 $\mathsf{ATL}^*\mathsf{K}$ path formula ψ as an LTL formula $lin(\psi)$ over a fresh set of atoms $max(\psi)$, the maximal history subformulas of ψ. This translation of $\mathsf{ATL}^*\mathsf{K}$ path formulas to LTL formulas does not make use of the assumption that the iCGS S only has public actions, and it is analogous to the translation of ATL^* (or CTL^*) path formulas to LTL formulas over maximal state subformulas [23, 1]. Specifically, let $max(\psi)$ be the set of history subformulas of ψ that are maximal, i.e., a history formula $\varphi \in max(\psi)$ iff it occurs in ψ and that occurence is not a subformula of any other occurence of a history subformula of ψ. If ψ 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 φ in ψ is

replaced by $\boxed{\varphi}$. For example, consider $\psi = (p \cup \langle\!\langle A \rangle\!\rangle K_a p) \vee X \neg C_A p$. The history subformulas of ψ are $\{p, \langle\!\langle A \rangle\!\rangle K_a p, K_a p, \neg C_A p, C_A p\}$. Then $max(\psi) = \{p, \langle\!\langle A \rangle\!\rangle K_a p, \neg C_A p\}$.[1] Thus $lin(\psi)$ is the LTL formula $\left(\boxed{p} \cup \boxed{\langle\!\langle A \rangle\!\rangle K_a p} \right) \vee X \boxed{\neg C_A p}$ 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 ψ is an LTL formula.

Construction of $M^s_{\langle\!\langle A \rangle\!\rangle \psi}$ *for an* LTL *formula* ψ. We will build a DPW $E_{\psi,s}$ over ACT that accepts $\alpha \in$ ACT^ω iff $(S, \mu(s, \alpha)) \models \psi$. 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 σ_A such that for every $\alpha \in \mathsf{ACT}^\omega$, if $\mu(s, u \cdot \alpha) \in \mathsf{out}(S, \mu(s,u), \sigma_A)$ then α is accepted by $E_{\psi,s}$ starting from state q (and thus $(S, \mu(s,u)) \models \langle\!\langle A \rangle\!\rangle \psi$, 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 ψ. 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 ψ can be converted into a DPW D_ψ over alphabet 2^{AP} that accepts all word models of that formula. Let Q_ψ 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_ψ 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^ps). 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_ψ of the state q, and $\langle q, f \rangle$ is an initial state if q is an initial state in D_ψ. The following is straightforward to prove:

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

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 σ of agent a by the ACT-branching tree $T_\sigma : \mathsf{ACT}^* \to (\mathrm{Act}_a)^{S_0}$ where, for $u \in \mathsf{ACT}^*$, $T_\sigma(u)(t) = \sigma(\mu(t,u))$. By Lemma 1, σ is uniform iff T_σ satisfies the property

$$\mu(t,u) \equiv_a \mu(t',u) \Rightarrow T_\sigma(u)(t) = T_\sigma(u)(t') \tag{1}$$

Lemma 4. *The set of encodings* T_σ *of uniform strategies* σ *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 $T_\sigma(u)(t) = T_\sigma(u)(t')$. $\square$

Encode a set of uniform strategies σ_A (for $A \subseteq \mathsf{Ag}$) as the convolution T_{σ_A} of their individual encodings, i.e., $T_{\sigma_A} : \mathsf{ACT}^* \to \prod_{a \in A} (\mathrm{Act}_a)^{S_0}$ where $T_{\sigma_A}(u)(a) = T_{\sigma_a}(u)$. Running the automata for σ_a in parallel yields:

Lemma 5. *For every* $A \subseteq \mathsf{Ag}$, *the set of encodings of joint uniform strategies* σ_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_{σ_A} iff for every $\alpha \in \mathsf{ACT}^\omega$, if $\mu(s, \alpha)$ is consistent with σ_A then α is accepted by $E_{\psi,s}$ starting from q.

---

[1] Note that although p has a non-maximal occurence in ψ, it is included in $max(\psi)$ since it has at least one occurence which is maximal, i.e., on the left-side of $\cup$.

The DPT $F_{\psi,s,q}$ works as follows: it reads T_{σ_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 σ_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 σ_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 $M_{\langle\langle A \rangle\rangle \psi}^{s}$. Finally, define the state q of $M_{\langle\langle A \rangle\rangle \psi}^{s}$ to be a final state iff the DPT $F_{\psi,s,q}$ is non-empty (a decidable condition).

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

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

3.1   Complexity

In this section we analyse the complexity of our decision procedure for a fixed number of agents. atoms. 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 the deterministic finite-word automaton (DFW) M_{φ}^{s} built in the proof of Theorem 5, for each case:

1. Atomic: $||p|| = O(1)$ for $p \in AP$.
2. Negation: $||\neg\varphi|| = ||\varphi||$.
3. Conjunction: $||\varphi \wedge \varphi'|| = ||\varphi|| \times ||\varphi'||$.
4. Epistemic: $||K_a\varphi|| = (2 \times |S| \times ||\varphi||)^{|S|}$, and the same for $||C_A\varphi||$ and $||D_A\varphi||$.
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 ψ, $\tilde{\psi}$ is the largest history subformula of ψ, and $AP(\cdot)$ is the set of atomic predicates occuring in its argument.

The last case requires some explanation. The deterministic parity tree automaton (DPT) accepting the set of all joint-strategies σ_A has size $O(2^{|S|^2})$, and has two colours. The deterministic parity word automaton (DPW) D_ψ has double-exponentially many states and single-exponentially many colours (in the size of $lin(\psi)$) [39, 36]. 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_{φ}^{s}. 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 $n^{O(m)}$ [20]. Thus, the time for constructing M_{φ}^{s} 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_{φ}^{s} 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 ε is accepted by this automaton has no additional cost. Thus, our algorithm runs in 2EXPTIME.

Lower-Bound. To prove the lower bound in Theorem 5 we reduce from a known 2EXPTIME-hard problem, i.e., model checking a CGS with $\mathrm{Ag} = \{a, b\}$ against a formula of the form $\langle\!\langle \{a\} \rangle\!\rangle \psi$ for an LTL formula ψ [34, 35]. One can translate a two-player CGS S into a polynomially larger CGS S' with public actions such that $S \models \varphi$ iff $S' \models \varphi$. Indeed, the agents, actions, and atoms are the same, $S' = S \times (\mathrm{ACT} \cup \{\varepsilon\})$, $S'_0 = S_0 \times \{\varepsilon\}$, $\delta'((s, d), d') = (\delta(s, d'), d')$, and $\lambda'(p) = \{(s, d) : s \in \lambda(p)\}$ (note that because S has two-players, $|\mathrm{ACT}| = |\mathrm{Act}_1| \times |\mathrm{Act}_2|$, and thus $|S'|$ is polynomial in the size of S).

4 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 [7]. Specifically, we considered a rich formal language to express complex strategic and epistemic properties of agents in MAS. This is the extension ATL*K 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 the objective interpretation of ATL modalities (the results also hold if we use the subjective semantics). 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 [28, 31] 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) [16, 17, 29, 32]. 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 [40]. 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 $K_a \varphi$ 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 and to implement them in an extension of the MCMAS tool for MAS verification [27].

References

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

[2] F. Belardinelli, A. Lomuscio, A. Murano & S. Rubin (2017): *Verification of Multi-agent Systems with Imperfect Information and Public Actions.* In: *AAMAS'17*, pp. 1268–1276.

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

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

[5] E. M. Clarke, O. Grumberg & D. A. Peled (1999): *Model Checking.* MIT Press.

[6] C. Dima, C. Enea & D. Guelev (2010): *Model-Checking an Alternating-time Temporal Logic with Knowledge, Imperfect Information, Perfect Recall and Communicating Coalitions.* In: *GandALF 2010*, *EPTCS* 25, pp. 103–117.

[7] Catalin Dima & Ferucio Laurentiu Tiplea (2011): *Model-checking ATL under Imperfect Information and Perfect Recall Semantics is Undecidable.* CoRR abs/1102.4225. Available at `http://arxiv.org/abs/1102.4225`.

[8] H. van Ditmarsch, W. van der Hoek & B. Kooi (2007): *Dynamic Epistemic Logic.* Synthese Library 337, Springer.

[9] H. van Ditmarsch & B. Kooi (2015): *One Hundred Prisoners and a Light Bulb.* Copernicus, doi:10.1007/978-3-319-16694-0_9.

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

[11] Kominis F & H. Geffner (2015): *Beliefs In Multiagent Planning: From One Agent to Many.* In: *ICAPS'15*, pp. 147–155.

[12] R. Fagin, J.Y. Halpern, Y. Moses & M.Y. Vardi (1995): *Reasoning about Knowledge.* MIT Press.

[13] B. Finkbeiner & S. Schewe (2005): *Uniform Distributed Synthesis.* In: *LICS'05*, IEEE Computer Society, pp. 321–330.

[14] Valentin Goranko & Wojciech Jamroga (2004): *Comparing Semantics of Logics for Multi-Agent Systems.* Synthese 139(2), pp. 241–280, doi:10.1023/B:SYNT.0000024915.66183.d1. Available at `http://dx.doi.org/10.1023/B%3ASYNT.0000024915.66183.d1`.

[15] Dimitar P. Guelev, Catalin Dima & Constantin Enea (2011): *An alternating-time temporal logic with knowledge, perfect recall and past: axiomatisation and model-checking.* Journal of Applied Non-Classical Logics 21(1), pp. 93–131, doi:10.3166/jancl.21.93-131. Available at `http://dx.doi.org/10.3166/jancl.21.93-131`.

[16] J. Gutierrez, P. Harrenstein & M. Wooldridge (2014): *Reasoning about Equilibria in Game-Like Concurrent Systems.* In: *KR'14*, AAAI.

[17] J. Gutierrez, P. Harrenstein & M. Wooldridge (2015): *Expresiveness and Complexity Results for Strategic Reasoning.* In: *CONCUR'15*, LIPIcs 42.

[18] W. van der Hoek & M. Wooldridge (2003): *Cooperation, Knowledge, and Time: Alternating-Time Temporal Epistemic Logic and its Applications.* Studia Logica 75(1), pp. 125–157.

[19] W. Jamroga & W. van der Hoek (2004): *Agents that Know How to Play.* Fundamenta Informaticae 62, pp. 1–35.

[20] M. Jurdzinski (2000): *Small Progress Measures for Solving Parity Games.* In: *STACS'00*, LNCS 1770, Springer, pp. 290–301.

[21] Piotr Kazmierczak, Thomas Ågotnes & Wojciech Jamroga (2014): *Multi-agency Is Coordination and (Limited) Communication*. In: *PRIMA'14*, LNCS 8861, Springer, pp. 91–106.

[22] O. Kupferman & M.Y. Vardi (2001): *Synthesizing Distributed Systems*. In: *LICS'01*, IEEE Computer Society, pp. 389–398.

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

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

[25] François Laroussinie, Nicolas Markey & Arnaud Sangnier (2015): *ATLsc with partial observation*. In: *GandALF 2015*, EPTCS 193, pp. 43–57.

[26] A. Lomuscio, R. van der Meyden & M. Ryan (2000): *Knowledge in Multi-Agent Systems: Initial Configurations and Broadcast*. ACM Trans. Comp. Log. 1(2), pp. 246–282.

[27] A. Lomuscio, H. Qu & F. Raimondi (2017): *MCMAS: A Model Checker for the Verification of Multi-Agent Systems*. Software Tools for Technology Transfer 19(1), pp. 9–30.

[28] Alessio R. Lomuscio, Ron van der Meyden & Mark Ryan (2000): *Knowledge in Multiagent Systems: Initial Configurations and Broadcast*. ACM Trans. Comput. Logic 1(2), pp. 247–284, doi:10.1145/359496.359527. Available at `http://doi.acm.org/10.1145/359496.359527`.

[29] A.D.C. Lopes, F. Laroussinie & N. Markey (2010): ATL *with Strategy Contexts: Expressiveness and Model Checking*. In: *FSTTCS'10*, LIPIcs 8, Leibniz-Zentrum fuer Informatik, pp. 120–132.

[30] R. van der Meyden & H. Shilov (1999): *Model Checking Knowledge and Time in Systems with Perfect Recall*. In: *FST&TCS'99*, LNCS 1738, Springer, pp. 432–445.

[31] R. van der Meyden & T. Wilke (2005): *Synthesis of Distributed Systems from Knowledge-Based Specifications*. In: *CONCUR'05*, LNCS 3653, Springer, pp. 562–576.

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

[33] H. Penny Nii (1986): *Blackboard Systems, Part One: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures*. AI Magazine 7(2), pp. 38–53. Available at `http://www.aaai.org/ojs/index.php/aimagazine/article/view/537`.

[34] A. Pnueli & R. Rosner (1990): *Distributed Reactive Systems Are Hard to Synthesize*. In: *FOCS'90*, IEEE Computer Society, pp. 746–757.

[35] R. Rosner (1992): *Modular Synthesis of Reactive Systems*. Ph.D. thesis, Weizmann Institute of Science, Rehovot, Israel.

[36] Shmuel Safra (1989): *Complexity of Automata on Infinite Objects*. Ph.D. thesis, Weizmann Institute of Science, Rehovot, Israel.

[37] Sven Schewe & Bernd Finkbeiner (2007): *Distributed Synthesis for Alternating-Time Logics*. In Kedar S. Namjoshi, Tomohiro Yoneda, Teruo Higashino & Yoshio Okamura, editors: *ATVA'07*, Springer, pp. 268–283, doi:10.1007/978-3-540-75596-8_20. Available at `http://dx.doi.org/10.1007/978-3-540-75596-8_20`.

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

[39] Moshe Y. Vardi & Pierre Wolper (1994): *Reasoning About Infinite Computations*. Information and Computation 115(1), pp. 1–37.

[40] Steen Vester (2013): *Alternating-time temporal logic with finite-memory strategies*. In: *GandALF'13*, EPTCS 119, pp. 194–207.