Towards Verification of Pushdown Epistemic Game Structures

Abstract

In this paper, we investigate the problem of verifying pushdown multi-agent systems with imperfect information. We introduce *pushdown epistemic game structures* (PEGSs), an extension of pushdown game structures with *epistemic accessibility relations* (EARs), as the formal model. For the specification, we consider extensions of alternating-time temporal logics with epistemic modalities: ATEL and ATEL*. We focus on strategies with imperfect recall. We show that size-preserving EARs will render the model checking problem undecidable. We then propose *regular* EARs, and provide automata-theoretic model checking algorithms with matching low bounds, i.e., EXPTIME-complete for ATEL and 2EXPTIME-complete for ATEL*.

1 Introduction

Model checking is a well-studied method for automatic formal verification of complex systems, and has been successfully applied to verify communication protocols, hardware designs and software, etc [Clarke *et al.*, 2001]. Recently, it has been extended to verify *multi-agent systems* (MASs). As a model of *finite-state* MASs, Alur et al. proposed *concurrent game structures* (CGSs), whilst alternating-time temporal logics (ATL, ATL*) and alternating-time μ-calculus (AMC) are employed as specification languages, for which model checking algorithms were also provided [Alur *et al.*, 1997; 2002]. Since then, a number of model checking algorithms for MASs have been studied for various models and logics, such as *strategy logic* (SL) and its fragments [Cermák *et al.*, 2015; Mogavero *et al.*, 2013; 2014b; 2014a].

CGSs are usually determined by *Interpreted Systems* which are constructed via a Mealy-type or Moore-type synchronous composition of agents' local transition systems [Fagin *et al.*, 1995; Lomuscio and Raimondi, 2006]. In the literature, the local transition system of each agent is usually of *finite* state (as,e.g., a Kripke structure), yielding a finite-state CGS only via the synchronous composition. However, in practice often there are scenarios of interest where agents *cannot* be represented by a finite-state system (cf. Section 3), but can be rightly modeled by a *pushdown* system. Hence, it would be of great interest to study verification problems on the CGS obtained by a synchronous composition of local *pushdown* systems. Unfortunately, the verification of even the simplest property (i.e., the reachability) for such a model is undecidable. To see this, one can easily reduce from the emptiness problem of the intersection of two pushdown automata which is known to be undecidable. To gain decidability while still capturing many interesting practical cases, *pushdown game structures* (PGSs) were proposed and investigated [Murano and Perelli, 2015; Chen *et al.*, 2016a; 2016b]. In PGSs agents do not posses their own local stacks, but can be seen as sharing a global stack. As the stack is unbounded, it represents a class of *infinite-state* MASs, a proper extension of the finite-state MASs. PGSs allow, among others, modeling of memory of agents, which is of particular importance in MASs.

On the logic side, one considers alternating-time temporal *epistemic* logics (ATEL, ATEL*) [van der Hoek and Wooldridge, 2002; Jamroga, 2003; Schobbens, 2004; Jamroga and Dix, 2006; Pilecki *et al.*, 2014], alternating-time *epistemic* μ-calculus (AEMC) [Bulling and Jamroga, 2011], and SLK [Cermák, 2015], which are respectively extensions of ATL, ATL*, AMC and SL with epistemic modalities for representing knowledge of individual agents, as well as "everyone knows" and common knowledge [Fagin *et al.*, 1995]. These logics are usually interpreted over *finite-state* concurrent *epistemic* game structures, which are an extension of CGSs with **epistemic accessibility relations** (EARs), giving rise to a model for representing finite-state MASs with *imperfect information*. Assuming agents only access imperfect information arises naturally in various real-world scenarios, typically in sensor networks, security, robotics, etc. In addition, the extension of logics with epistemic modalities allows one to succinctly express a range of (un)desirable properties of MASs, and has found a wide range of applications in AI, particularly for reasoning about MASs [Fagin *et al.*, 1995; van der Hoek and Wooldridge, 2003].

This paper investigates model checking problems for ATEL and ATEL* on *infinite-state* MASs with *imperfect information*. To this end, we propose *pushdown epistemic game structures* (PEGSs), an extension of PGSs with EARs, as a mathematical model for infinite-state MASs with imperfect information. To the best of our knowledge, analogous models have not been considered.

Model checking PEGSs depends crucially on how EARs are defined. A commonly adopted one, called *size-preserving* EARs, was introduced in [Aminof *et al.*, 2013], where two configurations are deemed to be indistinguishable if the two stack contents are of the same size, in addition, neither the pair of control states nor pairs of stack symbols in the same position of the two stack contents, are distinguishable. While this sounds to be a very natural definition, we show, unfortunately, that the model checking problems for ATEL and ATEL* on PEGSs are undecidable in general, even when restricted to imperfect recall strategies. This result suggests that alternative definitions of EARs are needed.

As a solution, we propose EARs that are *regular* and *simple*. Simple EARs are defined over control states of PEGSs and the top symbol of the stack, while regular EARs are simple EARs extended with a finite set of deterministic finite-state automata (DFA), one for each agent, where the states of each DFA divide the set of stack contents into finitely many equivalence classes. To obtain model checking algorithms, we first present an reduction from PEGSs with regular EARs to PEGSs with simple ones. We then provide automata-theoretic algorithms that solve the model checking problem for PEGSs with simple EARs. The algorithm runs in EXPTIME for ATEL and 2EXPTIME for ATEL*, and we show they are optimal by giving matching lower bounds.

These results suggest that adding appropriate EARs and modalities, which makes model and specification more expressive and succinct, does not bring computational overhead to the associated verification problems.

2 Pushdown Epistemic Game Structures

We fix a countable set **AP** of *atomic propositions* (also called observations). Let $[k]$ denote the set $\{1, ..., k\}$ for some natural number $k \in \mathbb{N}$.

Definition 1. *A* pushdown epistemic game structure *(PEGS) is a tuple* $\mathcal{P} = (\mathsf{Ag}, \mathsf{Ac}, P, \Gamma, \Delta, \lambda, \{\sim_i \mid i \in \mathsf{Ag}\})$*, where*

- $\mathsf{Ag} = \{1, ..., n\}$ *is a finite set of* agents *(a.k.a. players);*
- Ac *is a finite set of* actions *made by agents; we further define* $\mathcal{D} = \mathsf{Ac}^n$ *to be the set of* decisions $\mathbf{d} = \langle a_1, ..., a_n\rangle$ *such that* $\forall i \in [n]$, $\mathbf{d}(i) := a_i \in \mathsf{Ac}$*;*
- P *is a finite set of* control states*;*
- Γ *is a finite* stack alphabet*, which includes a special symbol* $\perp$ *to denote the bottom of the stack;*
- $\Delta : P \times \Gamma \times \mathcal{D} \to P \times \Gamma^*$ *is a transition function such that, for each* $(p, \gamma, \mathbf{d}) \in P \times \Gamma \times \mathcal{D}$*, if* $\gamma = \perp$*, then* $\delta(p, \gamma, \mathbf{d}) = (p', \omega\perp)$ *for some* $p' \in P$ *and* $\omega \in \Gamma^*$*; if* $\gamma \neq \perp$*, then* $\delta(p, \gamma, \mathbf{d}) = (p', \omega)$ *for some* $p \in P$ *and* $\omega \in (\Gamma \setminus \{\perp\})^*$ [1]*;*
- $\lambda : P \times (\Gamma^*\perp) \to 2^{\mathbf{AP}}$ *is a valuation that assigns to each configuration (i.e., an element of* $P \times (\Gamma^*\perp)$*) a set of atomic propositions (i.e., observations);*
- $\sim_i \subseteq (P\times(\Gamma^*\perp))\times(P\times(\Gamma^*\perp))$ *is an* epistemic accessibility relation *(EAR) which is an equivalence relation.*

A *concurrent epistemic game structure* (CEGS) is a PEGS where the stack is not used. We can define a CEGS as $\mathcal{P} = (\mathsf{Ag}, \mathsf{Ac}, P, \Delta, \lambda, \{\sim_i \mid i \in \mathsf{Ag}\})$, where $\Delta : P \times \mathcal{D} \to P$. Accordingly, λ and $\sim_i$ are over P solely. A *pushdown game structure* (PGS) is a PEGS $\mathcal{P} = (\mathsf{Ag}, \mathsf{Ac}, P, \Gamma, \Delta, \lambda, \{\sim_i \mid i \in \mathsf{Ag}\})$ in which $\sim_i$ is an identity for every agent $i \in \mathsf{Ag}$. Therefore, a PGS $\mathcal{P}$ is usually denoted as $(\mathsf{Ag}, \mathsf{Ac}, P, \Gamma, \Delta, \lambda)$. A PGS is a pushdown system (PDS) if $|\mathsf{Ag}| = 1$.

A *configuration* of the PEGS $\mathcal{P}$ is a pair $\langle p, \omega\rangle$, where $p \in P$ and $\omega \in \Gamma^*\perp$. We write $C_{\mathcal{P}}$ to denote the set of configurations of $\mathcal{P}$. For every $(p, \gamma, \mathbf{d}) \in P \times \Gamma \times \mathcal{D}$ such that $\Delta(p, \gamma, \mathbf{d}) = (p', \omega)$, we write $\langle p, \gamma\rangle \overset{\mathbf{d}}{\hookrightarrow}_{\mathcal{P}} \langle p', \omega\rangle$ instead.

The transition relation $\Longrightarrow_{\mathcal{P}}: C_{\mathcal{P}}\times\mathcal{D}\times C_{\mathcal{P}}$ of the PEGS $\mathcal{P}$ is defined as follows: for every $\gamma\omega' \in C_{\mathcal{P}}$, if $\langle p, \gamma\rangle \overset{\mathbf{d}}{\hookrightarrow}_{\mathcal{P}} \langle p', \omega\rangle$, then $\langle p, \gamma\omega'\rangle \overset{\mathbf{d}}{\Longrightarrow}_{\mathcal{P}} \langle p', \omega\omega'\rangle$. Intuitively, if the PEGS $\mathcal{P}$ is at the configuration $\langle p, \gamma\omega'\rangle$, by making the decision $\mathbf{d}$, $\mathcal{P}$ moves from the control state p to the control state p', pops γ from the stack and then pushes ω into the stack.

Tracks and Paths. A *track* (resp. *path*) in the PEGS $\mathcal{P}$ is a *finite* (resp. *infinite*) sequence π of configurations $c_0...c_m$ (resp. $c_0c_1...$) such that for every $i : 0 \leq i < m$ (resp. $i \geq 0$), $c_i \overset{\mathbf{d}}{\Longrightarrow}_{\mathcal{P}} c_{i+1}$ for some $\mathbf{d}$. Given a track $\pi = c_0...c_m$ (resp. path $\pi = c_0c_1...$), let $|\pi| = m$ (resp. $|\pi| = +\infty$), and for every $i : 0 \leq i \leq m$ (resp. $i \geq 0$), let π_i denote the configuration c_i, and $\pi_{\leq i}$ denote $c_0 \dots c_i$. Given two tracks π and π', π and π' are *distinguishable* for an agent $i \in \mathsf{Ag}$, denoted by $\pi \sim_i \pi'$, if for all $k : 0 \leq k \leq |\pi|$, $\pi_k \sim_i \pi'_k$. Let $\mathsf{Trks}_{\mathcal{P}} \subseteq C^+_{\mathcal{P}}$ denote the set of all tracks in $\mathcal{P}$, $\prod_{\mathcal{P}} \subseteq C^\omega_{\mathcal{P}}$ denote the set of all paths in $\mathcal{P}$, $\mathsf{Trks}_{\mathcal{P}}(c) = \{\pi \in \mathsf{Trks}_{\mathcal{P}} \mid \pi_0 = c\}$ and $\prod_{\mathcal{P}}(c) = \{\pi \in \prod_{\mathcal{P}} \mid \pi_0 = c\}$ respectively denote the set of all the tracks and paths starting from the configuration c.

Strategies. Intuitively a *strategy* of an agent $i \in \mathsf{Ag}$ specifies what i plans to do in each situation. In the literature, there are four types of strategies [Schobbens, 2004; Bulling and Jamroga, 2011] defined as follows: where **i** (resp. **I**) denotes imperfect (resp. perfect) information and **r** (resp. **R**) denotes imperfect (resp. perfect) recall,

- **Ir** strategy is a function $\theta_i : C_{\mathcal{P}} \to \mathsf{Ac}$, i.e., the action made by the agent i depends on the current configuration;
- **IR** strategy is a function $\theta_i : \mathsf{Trks}_{\mathcal{P}} \to \mathsf{Ac}$, i.e., the action made by the agent i depends on the history, i.e. the sequence of configurations visited before;
- **ir** strategy is a function $\theta_i : C_{\mathcal{P}} \to \mathsf{Ac}$ such that for all configurations $c, c' \in C_{\mathcal{P}}$, if $c \sim_i c'$, then $\theta_i(c) = \theta_i(c')$, i.e., the agent i has to make the same action at the configurations that are indistinguishable from each other;
- **iR** strategy is a function $\theta_i : \mathsf{Trks}_{\mathcal{P}} \to \mathsf{Ac}$ such that for all tracks $\pi, \pi' \in \mathsf{Trks}_{\mathcal{P}}$, if $\pi \sim_i \pi'$, then $\theta_i(\pi) = \theta_i(\pi')$, i.e., the agent i has to make the same action on the tracks that are indistinguishable from each other.

[1] One may notice that, in the definition of PEGSs, Δ is defined as a *complete* function $P \times \Gamma \times \mathcal{D} \to P \times \Gamma^*$, meaning that all actions are available to each agent. This does not restrict the expressiveness of PEGSs, as we can easily add the transitions to some additional sink state to simulate the situation that some actions are unavailable to some agents.

Let Θ^σ for $\sigma \in \{\mathbf{Ir}, \mathbf{IR}, \mathbf{ir}, \mathbf{iR}\}$ denote the set of all σ-strategies. Given a set of agents $A \subseteq \mathsf{Ag}$, a *collective σ-strategy* of A is a function $\upsilon_A : A \to \Theta^\sigma$ that assigns to each agent $i \in A$ a σ-strategy. We write $\overline{A} = \mathsf{Ag} \setminus A$.

Outcomes. Let c be a configuration and υ_A be a collective σ-strategy for a set of agents A. A path π is *compatible* with respect to υ_A iff for every $k \geq 1$, there exists $\mathbf{d}_k \in \mathcal{D}$ such that $\pi_{k-1} \overset{\mathbf{d}_k}{\Longrightarrow}_{\mathcal{P}} \pi_k$ and $\mathbf{d}_k(i) = \upsilon_A(i)(\pi_{\leq k-1})$ for all $i \in A$. The *outcome starting from c with respect to υ_A*, denoted by $\mathsf{out}^\sigma(c, \upsilon_A)$, is defined as the set of all the paths that start from c and are compatible with respect to υ_A, which rules out infeasible paths with respect to the collective σ-strategy υ_A.

Epistemic accessibility relations (EARs). An EAR $\sim_i$ for $i \in \mathsf{Ag}$ over PEGSs is defined as an equivalence relation over configurations. As the set of configurations is infinite in general, we need to represent each $\sim_i$ *finitely*.

A very natural definition of EARs, called *size-preserving* EARs and considered in [Aminof *et al.*, 2013], is formulated as follows: for each $i \in \mathsf{Ag}$, there is an equivalence relation $\simeq_i \subseteq (P \times P) \cup (\Gamma \times \Gamma)$, which captures the indistinguishability of control states and stack symbols. For two configurations $c = \langle p, \gamma_1 ... \gamma_m \rangle$ and $c' = \langle p', \gamma'_1 ... \gamma'_{m'} \rangle$, $c \sim_i c'$ iff $m = m'$, $p \simeq_i p'$, and for every $j \in [m] = [m']$, $\gamma_j \simeq_i \gamma'_j$. It turns out that the model checking problem for logic ATEL/ATEL* (cf. Section 4) is undecidable under this type of EARs, even with imperfect recall (cf. Theorem 3). To gain decidability, in this paper, we consider *regular EARs* and a special case thereof, i.e. *simple* EARs. We remark that regular EARs align to the regular valuations (see later of this section) of atomic propositions, can been seen as approximations of EARs, and turn out to be useful in practice.

An EAR $\sim_i$ is *regular* if there is an equivalence relation $\approx_i$ over $P \times \Gamma$ and a complete deterministic finite-state automaton[2] (DFA) $\mathcal{A}_i = (S_i, \Gamma, \delta_i, s_{i,0})$ such that for every $\langle p, \gamma\omega \rangle, \langle p_1, \gamma_1\omega_1 \rangle \in C_{\mathcal{P}}$, $\langle p, \gamma\omega \rangle \sim_i \langle p_1, \gamma_1\omega_1 \rangle$ iff $(p, \gamma) \approx_i (p_1, \gamma_1)$ and $\delta_i^*(s_{i,0}, \omega^R) = \delta_i^*(s_{i,0}, \omega_1^R)$, where δ_i^* denotes the reflexive and transitive closure of δ_i, and ω^R, ω_1^R denote the reverse of ω, ω_1 (recall that the rightmost symbol of ω corresponds to the bottom symbol of the stack). Intuitively, two words ω, ω_1 which record the stack content (excluding the top), are equivalent with respect to $\sim_i$ if the two runs of $\mathcal{A}_i$ on ω^R and ω_1^R respectively reach the same state. Note that the purpose of DFA $\mathcal{A}_i$ is to partition Γ^* into finitely many equivalence classes, hence we do *not* introduce the accepting states. A regular EAR is *simple* if for every word $\omega, \omega_1 \in \Gamma^*$, $\delta_i^*(s_{i,0}, \omega^R) = \delta_i^*(s_{i,0}, \omega_1^R)$, that is, $\mathcal{A}_i$ contains only one state.

Given a set of agents $A \subseteq \mathsf{Ag}$, let $\sim_A^E$ denote $\bigcup_{i \in A} \sim_i$, and $\sim_A^C$ denote the transitive closure of $\sim_A^E$.

Alternating Multi-Automata. To represent potentially infinite sets of configurations finitely, we use alternating multi-automata (AMA) as the "data structure" of the model checking algorithms.

Definition 2. *[Bouajjani* et al.*, 1997] Given a PDS $\mathcal{P} = (Ag, Ac, P, \Gamma, \Delta, \lambda)$, an AMA is a tuple $\mathcal{M} = (S, \Gamma, \delta, I, S_f)$, where S is a finite set of states such that $P \subseteq S$, Γ is the input alphabet, $\delta \subseteq S \times \Gamma \times 2^S$ is a transition relation, $I \subseteq S$ is a finite set of initial states, $S_f \subseteq S$ is a finite set of final states.*

[2]"complete" means that $\delta(q, \gamma)$ is defined for each $(q, \gamma) \in Q \times \Gamma$.

If $(s, \gamma, \{s_1, ..., s_m\}) \in \delta$, we will write $s \xrightarrow{\gamma} \{s_1, ..., s_m\}$ instead. We define the relation $\longrightarrow_\delta \subseteq S \times \Gamma^* \times 2^S$ as the least relation such that the following conditions hold: (1) $s \xrightarrow{\epsilon}_\delta \{s\}$, for every $s \in S$; (2) $s \xrightarrow{\gamma\omega}_\delta \bigcup_{i \in [m]} S_i$, if $s \xrightarrow{\gamma} \{s_1, ..., s_m\}$ and $s_i \xrightarrow{\omega}_\delta S_i$ for every $i \in [m]$.

$\mathcal{M}$ *accepts* a configuration $\langle p, \omega \rangle$ if $p \in I$ and there exists $S' \subseteq S_f$ such that $p \xrightarrow{\omega}_\delta S'$. Let $\mathcal{L}(\mathcal{M})$ denote the set of all configurations accepted by $\mathcal{M}$. A set $C \subseteq C_{\mathcal{P}}$ is *regular* iff some AMA $\mathcal{M}$ exactly recognizes C.

Proposition 1. *[Bouajjani* et al.*, 1997] The membership problem of AMAs can be decided in polynomial time. AMAs are closed under all Boolean operations.*

Regular valuations. The model checking problem for PDSs (hence for PEGSs as well) with general valuations λ, e.g., defined by a function l which assigns to each atomic proposition a context free language, is undecidable [Kupferman *et al.* , 2002]. To gain decidability, we consider valuations specified by a function $l : \mathbf{AP} \to 2^{C_{\mathcal{P}}}$ such that for every $q \in \mathbf{AP}$, $l(q)$ is a *regular* set of configurations. This is usually referred to as a *regular valuation* [Esparza *et al.*, 2003]. The function l can be lifted to the valuation $\lambda_l : P \times \Gamma^* \to 2^{\mathbf{AP}}$: for every $c \in C_{\mathcal{P}}$, $\lambda_l(c) = \{q \in \mathbf{AP} \mid c \in l(q)\}$. A *simple valuation* is a regular valuation $l : \mathbf{AP} \to 2^{C_{\mathcal{P}}}$ such that for every $q \in \mathbf{AP}$, $p \in P, \gamma \in \Gamma$, and $\omega, \omega' \in \Gamma^*$, it holds that $\langle p, \gamma\omega \rangle \in l(q)$ iff $\langle p, \gamma\omega' \rangle \in l(q)$. From now on, we assume that regular valuations are given as AMAs. For a PEGS $\mathcal{P}$ with regular valuations, we use $|\mathcal{P}|$ to denote the number of control states of $\mathcal{P}$ and the numbers of states of these AMAs.

3 Abilities of Stack

Theoretically, pushdown game structures (PGSs) and PEGSs respectively are more expressive than (finite-state) CGSs and CEGSs. In practice, the stack could be used as a last-in-first-out (LIFO) global memory, as shown in [Murano and Perelli, 2015] where each agent is represented by a local finite-state transition system and can store into or load from the stack using push and pop. In this section, we present another potential application of PGSs/PEGSs.

As mentioned in Section 1, game structures are usually constructed by taking a product of agents' local transition systems. Unfortunately, allowing each agent to have a local stack would immediately lead to undecidability even for the reachability problem with two agents. We show that, by a proper restriction on the local stack operations (i.e., push and pop) of agents, one can obtain a game structure representable by PGSs/PEGSs. For example, all agents are restricted to *synchronously* push (resp. pop) same number of local stack symbols into (resp. from) their local stacks. In this case, a PGS/PEGS can be constructed by taking a product of local states and local stacks of all agents, in particular, the global stack symbol is the vector of its local counterparts.

In many practical scenarios, an agent needs to keep track of all the local history information she had along a run. Consequently, an agent's local state will "grow", which requires an *infinite-state* transition system to model. We can resort to pushdown systems where the stack is used to keep track of the history information the agent had. It is easy to see that for this purpose, all agents indeed synchronously push (resp. pop) local stack symbols into (resp. from) their local stacks at each step. As a result, the whole game structure can be modeled as a PGS or PEGS. Explicitly keeping track of the history enhances the capabilities of agents significantly and is practically relevant. For instance, in multi-agent systems for e-commerce [Tsai *et al.*, 2005] or automated trading [Ash, 2004], when a transaction is canceled, the agents must be able to synchronously rollback and recover the previous saved checkpoint. Verification of this class of MASs would be hard, if not impossible, without introducing PGS/PEGS.

On a different matter, using regular valuations, observations (i.e., atomic propositions) can range over not only the current states, but also the history information. For example, one can express properties like "the atomic proposition q holds iff the agent 1 is at local state s_1 and the agent 2 has visited the local state s_2 twice, but the agent 3 has not visited the state s_3 yet."

Example 1. *Consider a system with two processes P_1 and P_2, each of them has a Boolean variable x_i. If x_i=*false*, P_i either changes it to* true *or leaves it unchanged. Otherwise, P_i leaves the value of x_i unchanged. Moreover, x_2 can be changed from* false *to* true *only if x_1=*true *or is simultaneously set to* true*. The system enters a fault state if one of the following conditions occurs: (1) x_1=x_2=*false *and P_2 chooses to change the value of x_2 but P_1 leaves x_1 unchanged, or (2) x_1=*true*, x_2=*false *and P_1, P_2 choose to change the value of their variables. We model the system as a CGS $\mathcal{P} = (\mathsf{Ag} = \{1,2\}, \mathsf{Ac} = \{a_1, a_2\}, P, \Delta, \lambda)$: a_1 denotes change the value of the variable and a_2 leaves the value of the variable unchanged. $P = \{p_f, p_{00}, p_{10}, p_{11}\}$. p_f denotes the fault state, $p_{1,0}$ denotes x_1=*true *and x_2=*false*, other control states are defined accordingly. The values of x_i's are observable, we use two atomic propositions $q_i's$ to denote $x_i's$ and q_f to denote the fault such that $\lambda(q_1) = \{p_{10}, p_{11}\}$, $\lambda(q_2) = \{p_{11}\}$ and $\lambda(q_f) = \{p_f\}$. Δ is defined as*

$\Delta(p_{00}, \mathbf{d}_{11}) = p_{11}$	$\Delta(p_{00}, \mathbf{d}_{12}) = p_{10}$	$\Delta(p_{00}, \mathbf{d}_{22}) = p_{00}$
$\Delta(p_{00}, \mathbf{d}_{21}) = p_f$	$\Delta(p_{10}, \mathbf{d}_{21}) = p_{11}$	$\Delta(p_{10}, \mathbf{d}_{22}) = p_{10}$
$\Delta(p_{10}, \mathbf{d}_{11}) = p_f$	$\Delta(p_{10}, \mathbf{d}_{12}) = p_{11}$	$\Delta(p, \mathbf{d}) = p$

where $\mathbf{d} \in \mathcal{D}$, $\mathbf{d}_{ij} = \langle a_i, a_j\rangle$ and $p \in \{p_f, p_{11}\}$.

To make the maximal use of processes' reasoning capabilities, we can explicitly keep track the information she had along a run by modeling the system as the PGS $\mathcal{P}' = (\{1,2\}, \{a_1, a_2\}, P, \Gamma = P \cup \{\bot\}, \Delta', \lambda)$, where $\Delta'(p, \gamma, \mathbf{d}) = (p', p\gamma)$, for every $\gamma \in \Gamma$, $\mathbf{d} \in \mathcal{D}$, $p, p' \in P$ such that $\Delta(p, \mathbf{d}) = p'$.

To achieve fault tolerance in the system, once the system enters the fault state, the two processes will synchronously recover their values and synchronously rollback the tracked information. In this case, we could add a fresh action a_3 into the PGS model $\mathcal{P}'$ and extend Δ' by setting $\Delta'(p_f, \gamma, \langle a_3, a_3\rangle) = (\gamma, \epsilon)$. (Note that the top-stack symbol γ is a state.)

If each process P_i can only see the value of x_i and its local history information, we can introduce two EARs $\sim_1$ and $\sim_2$ into $\mathcal{P}'$ such that for every $c = \langle p_{i_0^1 i_0^2}, p_{i_1^1 i_1^2} ... p_{i_m^1 i_m^2}\rangle, c' = \langle p_{j_0^1 j_0^2}, p_{j_1^1 j_1^2} ... p_{j_k^1 j_k^2}\rangle \in C_{\mathcal{P}'}$, $c \sim_1 c'$ (resp. $c \sim_2 c'$) iff $m = k$, and $i_r^1 = j_r^1$ (resp. $i_r^2 = j_r^2$), for every $r : 0 \le r \le m$. In this case, $\sim_1$ and $\sim_2$ are size-preserving EARs. If the actions made by the agent 1 only depend on the local history of the n latest step, then $c \sim_1 c'$ iff $\min(m, n) = \min(k, n)$ and $i_r^1 = j_r^1$ for every $r : 0 \le r \le \min(m, n)$. Obviously, this EAR $\sim_1$ can be characterised by a finite state automaton, thus a regular EAR.

4 ATEL and ATEL*

In this section, we recall the definition of alternating-time temporal epistemic logics: ATEL [van der Hoek and Wooldridge, 2002] and ATEL* [Jamroga, 2003], which were introduced for reasoning about knowledge and cooperation of agents in multi-agent systems. Informally, ATEL and ATLE* can be considered as extensions of ATL and ATL* respectively with *epistemic modalities* for representing knowledge. These include $\mathbf{K}_i$ for $i \in \mathsf{Ag}$ (agent i knows), $\mathbf{E}_A$ for $A \subseteq \mathsf{Ag}$ (every agent in A knows) and $\mathbf{C}_A$ (group modalities to characterise common knowledge).

4.1 ATEL$^*_\sigma$ (where $\sigma \in \{\mathbf{Ir}, \mathbf{IR}, \mathbf{ir}, \mathbf{iR}\}$)

Definition 3. *The syntax of ATEL$^*_\sigma$ is defined as follows, where ϕ denotes state formulae, ψ denotes path formulae,*

$$\begin{aligned}\phi ::= &\ q \mid \neg q \mid \phi \vee \phi \mid \phi \wedge \phi \mid \mathbf{K}_i\phi \mid \mathbf{E}_A\phi \mid \\ &\ \mathbf{C}_A\phi \mid \overline{\mathbf{K}}_i\phi \mid \overline{\mathbf{E}}_A\phi \mid \overline{\mathbf{C}}_A\phi \mid \langle A\rangle\psi \mid [A]\psi \\ \psi ::= &\ \phi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathbf{X}\,\psi \mid \mathbf{G}\,\psi \mid \psi\,\mathbf{U}\,\psi\end{aligned}$$

where $q \in \mathbf{AP}$, $i \in \mathsf{Ag}$ and $A \subseteq \mathsf{Ag}$.

We use $\mathbf{F}\,\psi$ to abbreviate *true* $\mathbf{U}\,\psi$. An LTL formula is an ATEL$^*_\sigma$ path formula ψ with ϕ being restricted to be atomic propositions and their negations.

The semantics of ATEL$^*_\sigma$ is defined over PEGSs. Let $\mathcal{P} = (\mathsf{Ag}, \mathsf{Ac}, P, \Gamma, \Delta, \lambda, \{\sim_i \mid i \in \mathsf{Ag}\})$ be a PEGS, ϕ be an ATEL$^*_\sigma$ state formula, $c \in C_{\mathcal{P}}$ be a configuration of $\mathcal{P}$, the satisfiability relation $\mathcal{P}, c \models_\sigma \phi$ is defined inductively on the structure of ϕ.

- $\mathcal{P}, c \models_\sigma q$, $\mathcal{P}, c \models_\sigma \neg q$, $\mathcal{P}, c \models_\sigma \phi_1 \vee \phi_2$ and $\mathcal{P}, c \models_\sigma \phi_1 \wedge \phi_2$ are defined in a standard way;
- $\mathcal{P}, c \models_\sigma \langle A\rangle\psi$ iff there exists a collective σ-strategy $\upsilon_A : A \to \Theta^\sigma$ s.t. for all paths $\pi \in \mathsf{out}^\sigma(c, \upsilon_A)$, $\mathcal{P}, \pi \models_\sigma \psi$;
- $\mathcal{P}, c \models_\sigma [A]\psi$ iff for all collective σ-strategies $\upsilon_A : A \to \Theta^\sigma$, there exists a path $\pi \in \mathsf{out}^\sigma(c, \upsilon_A)$, $\mathcal{P}, \pi \models_\sigma \psi$;
- $\mathcal{P}, c \models_\sigma \mathbf{K}_i\phi$ iff for all configurations $c' \in C_{\mathcal{P}}$ such that $c \sim_i c'$, $\mathcal{P}, c' \models_\sigma \phi$;
- $\mathcal{P}, c \models_\sigma \overline{\mathbf{K}}_i\phi$ iff there is a configuration $c' \in C_{\mathcal{P}}$ such that $c \sim_i c'$ and $\mathcal{P}, c' \models_\sigma \phi$;
- $\mathbf{E}_A\phi$, $\overline{\mathbf{E}}_A\phi$, $\mathbf{C}_A\phi$ and $\overline{\mathbf{C}}_A\phi$ are defined similar to $\mathbf{K}_i\phi$ and $\overline{\mathbf{K}}_i\phi$, using the relation $\sim_A^E$ and $\sim_A^C$.

The semantics of path formulae ψ is specified by a relation $\mathcal{P}, \pi \models_\sigma \psi$, where π is a path. Since the definition is essentially the one of LTL and standard, we refer the readers to, e.g. [Clarke *et al.*, 2001] for details. We denote by $\|\phi\|^\sigma_{\mathcal{P}} = \{c \in C_{\mathcal{P}} \mid \mathcal{P}, c \models_\sigma \phi\}$ the set of configurations satisfying ϕ. The *model checking problem* is to decide whether $c \in \|\phi\|^\sigma_{\mathcal{P}}$ for a given configuration c.

Example 2. *Recalling Example 1, the formula $\langle 1\rangle \mathbf{G}\neg q_f$ expresses that "1 has a strategy such that the fault will never happen whatever 2 does", the formula $\mathbf{K}_2\langle 1\rangle \mathbf{G}\neg q_f$ expresses that "2 knows that 1 has a strategy such that the fault will never happen whatever 2 does", and the formula $\mathbf{E}_{\{1,2\}}\langle 2\rangle true \mathbf{U}(q_1 \wedge q_2)$ expresses that "1, 2 know that 2 has a strategy such that both x_1 and x_2 will eventually be true".*

ATEL_σ is a syntactical fragment of ATEL^*_σ with restricted path formulae defined by the rule: $\psi ::= \mathbf{X}\,\phi \mid \mathbf{G}\,\phi \mid \phi\,\mathbf{U}\,\phi$, where ϕ are state formulae.

CTL and CTL* are special cases of ATL_σ and ATL^*_σ in which all the modalities $\langle A\rangle\psi$ and $[A]\psi$ satisfy that $A = \emptyset$, while the latter two are special cases of ATEL_σ and ATEL^*_σ in which no epistemic modalities occur.

An ATEL_σ (resp. ATEL^*_σ) formula ϕ is a *principal* if ϕ is in the form of $\langle A\rangle\psi$ or $[A]\psi$ such that ψ is an LTL formula. For instance, $\langle 1\rangle Fq$ is a principal formula, while neither $\langle 1\rangle F(q \wedge \langle 2\rangle Gq')$ nor $\langle 1\rangle F(K_2\, q)$ is.

Remark 1. *In [Bulling and Jamroga, 2011], the outcome of a configuration c with respect to a given collective σ-strategy υ_A is defined differently from that in this paper. More specifically, the outcome in [Bulling and Jamroga, 2011] corresponds to $\bigcup_{i\in A}\bigcup_{c\sim_i c'} \mathsf{out}^\sigma(c', \upsilon_A)$ in our notation. It is easy to see that for every $ATEL_\sigma$ or $ATEL^*_\sigma$ formula $\langle A\rangle\psi$ (resp. $[A]\psi$) and every configuration $c \in C_{\mathcal{P}}$, $c \in \|\langle A\rangle\psi\|^\sigma_{\mathcal{P}}$ (resp. $c \in \|[A]\psi\|^\sigma_{\mathcal{P}}$) in [Bulling and Jamroga, 2011] iff $c \in \|\mathbf{E}_A\langle A\rangle\psi\|^\sigma_{\mathcal{P}}$ (resp. $c \in \|\mathbf{E}_A[A]\psi\|^\sigma_{\mathcal{P}}$) in our notation. We decide to make the hidden epistemic modalities $\mathbf{E}_A$ explicit in this paper.*

The following results are known for model checking PEGSs with perfect information and perfect recall.

Theorem 1 ([Chen *et al.*, 2016a]). *The model checking problem on PEGSs is EXPTIME-complete for $ATEL_{\mathbf{IR}}$, and 3EXPTIME-complete for $ATEL^*_{\mathbf{IR}}$.*

5 ATEL and ATEL* Model Checking

We first recall the following undecidability result.

Theorem 2 ([Dima and Tiplea, 2011]). *The model checking problem of $ATL_{\mathbf{iR}}$ and $ATL^*_{\mathbf{iR}}$ on CEGSs is undecidable.*

In light of Theorem 1 and Theorem 2, in this section, we focus on the model checking problems for $\mathrm{ATEL}_{\mathbf{ir}}$/$\mathrm{ATEL}^*_{\mathbf{ir}}$. (Results for $\mathrm{ATEL}_{\mathbf{Ir}}$/$\mathrm{ATEL}^*_{\mathbf{Ir}}$ can be obtained as a special case when $\sim_i$ in the PEGS is an identity for each agent i.)

We observe that, when the stack is available, the histories in CEGSs can be stored into the stack, so that we can reduce from the model checking problem for $\mathrm{ATL}_{\mathbf{iR}}$ on CEGSs to the one for $\mathrm{ATL}_{\mathbf{ir}}$ on PEGSs. From Theorem 2, we deduce the following result.

Theorem 3. *The model checking problems for $ATL_{\mathbf{ir}}$/$ATL^*_{\mathbf{ir}}$ on PEGSs with size-preserving EARs are undecidable.*

Theorem 3 rules out model checking algorithms for $\mathrm{ATEL}_{\mathbf{ir}}$/$\mathrm{ATEL}^*_{\mathbf{ir}}$ when the PEGS is equipped with size-preserving EARs. As mentioned before, we therefore consider the case with regular or simple EARs. We first give a reduction from the former to the latter as, evidently, the latter problem is considerably simpler. The main idea of the reduction, which is inspired by the reduction of PDSs with regular valuations to PDSs with simple valuations in [Esparza *et al.*, 2003], is to store the states of DFAs representing the regular EARs into the stack.

5.1 Reduction from Regular to Simple EARs

Assume a PEGS $\mathcal{P} = (\mathsf{Ag}, \mathsf{Ac}, P, \Gamma, \Delta, \lambda, \{\sim_i \mid i \in \mathsf{Ag}\})$ with regular EARs such that, for each $i \in \mathsf{Ag}$, $\sim_i$ is given as the pair of $(\approx_i, \mathcal{A}_i)$, where $\approx_i \subseteq P \times \Gamma$ is an equivalence relation and $\mathcal{A}_i = (S_i, \Gamma, \delta_i, s_{i,0})$ is a DFA.

Let $\vec{\mathcal{A}} = (\vec{S}, \Gamma, \vec{\delta}, \vec{s_0})$ be the product automaton of $\mathcal{A}_i$'s for $i \in \mathsf{Ag}$. Intuitively, the PEGS $\mathcal{P}'$ with simple EARs to be constructed, stores the state obtained by running $\vec{\mathcal{A}}$ over the reverse of the partial stack content up to the current position (exclusive) into the stack, Formally, the PEGS $\mathcal{P}' = (\mathsf{Ag}, \mathsf{Ac}, P, \Gamma', \Delta', \lambda', \{\sim'_i \mid i \in \mathsf{Ag}\})$ where $\Gamma' = \Gamma \times \vec{S}$, and for each $i \in \mathsf{Ag}$, $\sim'_i$ is specified by an equivalence relation $\approx'_i$ on $P \times \Gamma'$ defined as follows: $(p, [\gamma, \vec{s}]) \approx'_i (p', [\gamma', \vec{s'}])$ iff $(p, \gamma) \approx_i (p', \gamma')$ and $\vec{s} = \vec{s'}$, in addition, Δ' is defined as follows: for every state $\vec{s} \in \vec{S}$,

1. for every $\langle p, \gamma\rangle \overset{\mathbf{d}}{\hookrightarrow}_{\mathcal{P}} \langle p', \epsilon\rangle$, $\langle p, [\gamma, \vec{s}]\rangle \overset{\mathbf{d}}{\hookrightarrow}_{\mathcal{P}'} \langle p', \epsilon\rangle$,
2. for every $\langle p, \gamma\rangle \overset{\mathbf{d}}{\hookrightarrow}_{\mathcal{P}} \langle p', \gamma_k \ldots \gamma_1\rangle$ with $k \geq 1$ and $\vec{\delta}(\vec{s_j}, \gamma_j) = \overrightarrow{s_{j+1}}$ for every $j : 1 \leq j \leq k-1$ (where $\vec{s_1} = \vec{s}$), then $\langle p, [\gamma, \vec{s}]\rangle \overset{\mathbf{d}}{\hookrightarrow}_{\mathcal{P}'} \langle p', [\gamma_k, \vec{s_k}] \ldots [\gamma_1, \vec{s_1}]\rangle$.

Finally, the valuation λ is adjusted accordingly to λ'.

Theorem 4. *The model checking problem of an $ATEL_{\mathbf{ir}}$ (resp. $ATEL^*_{\mathbf{ir}}$) formula ϕ on a PEGS $\mathcal{P}$, with stack alphabet Γ and regular EARs $\sim_i = (\approx_i, \mathcal{A}_i)$ for $i \in \mathsf{Ag}$, can be reduced to the model checking problem of ϕ on a PEGS $\mathcal{P}'$ with simple EARs $\sim'_i$, such that the state space of $\mathcal{P}'$ is the same as that of $\mathcal{P}$, and the stack alphabet of $\mathcal{P}'$ is $\Gamma \times \vec{S}$, where $\vec{S}$ is the state space of the product of $\mathcal{A}_i$'s for $i \in \mathsf{Ag}$.*

Theorem 4 allows us to focus on the model checking problem over PEGSs with simple EARs.

5.2 Model Checking $\mathrm{ATEL}_{\mathbf{ir}}$ and $\mathrm{ATEL}^*_{\mathbf{ir}}$ on PEGSs with Simple EARs

Assume an $\mathrm{ATEL}_{\mathbf{ir}}$ (resp. $\mathrm{ATEL}^*_{\mathbf{ir}}$) formula ϕ and a PEGS $\mathcal{P} = (\mathsf{Ag}, \mathsf{Ac}, P, \Gamma, \Delta, \lambda, \{\sim_i \mid i \in \mathsf{Ag}\})$ with regular valuations l such that $l(q)$ is given by an AMA $\mathcal{M}_q$ for each $q \in AP$ and $\sim_i$ is specified by an equivalence relation $\approx_i$ on $P \times \Gamma$ for $i \in \mathsf{Ag}$.

We propose automata-theoretic model checking algorithms for ϕ on $\mathcal{P}$. The idea of the algorithm is to construct, for each state subformula ϕ' of ϕ, an AMA $\mathcal{M}_{\phi'}$ to represent the set of configurations satisfying ϕ'. We will first illustrate the construction in case that ϕ' is a principal formula, then extend the construction to the more general case.

Principal formulae. In the following, we will illustrate the construction for principal formulae $\phi' = \langle A\rangle\psi'$, where ψ' is an LTL formula. The construction for principal formulae $\phi' = [A]\psi'$ is similar. We assume that each atomic proposition $q \in AP$ is associated with an AMA $\mathcal{M}_q$ to denote the set of configurations satisfying q.

Our approach will reduce the model checking problem on PEGSs to the problem of PDSs. Note that for $i \in A$, $\approx_i$ is defined over $P \times \Gamma$. It follows that the strategy of any agent $i \in A$ must respect $\approx_i$, i.e., for all $(p, \gamma\omega)$ and $(p', \gamma'\omega')$ with $(p, \gamma) \approx_i (p', \gamma')$, $\upsilon_i(p, \gamma\omega) = \upsilon_i(p', \gamma'\omega')$ for any **ir**-strategy υ_i of i. Therefore, any **ir**-strategy υ_i with respect to $\approx_i$ can be regarded as a function over $P \times \Gamma$ (instead of configurations of $\mathcal{P}$), i.e., $\upsilon_i : P \times \Gamma \to \mathsf{Ac}$.

For each $(p, \gamma) \in P \times \Gamma$, after applying a collective **ir**-strategy $\upsilon_A = (\upsilon_i(p, \gamma))_{i \in A}$ for A, we obtain a PDS $\mathcal{P}_{\upsilon_A} = (P, \Gamma, \Delta', \lambda)$, where $\Delta'(p, \gamma, \mathbf{d}') = \Delta(p, \gamma, \mathbf{d})$, for every $i \in A$ with $\mathbf{d}(i) = \upsilon_i(p, \gamma)$, $\mathbf{d}'$ is the action vector of the agents $\overline{A}$ in $\mathbf{d}$. Then, $\mathsf{out}^{\mathbf{ir}}(c, \upsilon_A) = \prod_{\mathcal{P}_{\upsilon_A}}(c)$, the set of all paths of $\mathcal{P}_{\upsilon_A}$ starting from c. It follows that $\mathcal{P}, c \models_{\mathbf{ir}} \langle A \rangle \psi'$ iff $\exists \upsilon_A$ such that $\mathcal{P}_{\upsilon_A}, c \models_{\mathbf{ir}} \langle \emptyset \rangle \psi'$, where $\langle \emptyset \rangle \psi'$ is a CTL (resp. CTL*) formula if $\langle A \rangle \psi'$ is an ATEL$_{\mathbf{ir}}$ (resp. ATEL$^*_{\mathbf{ir}}$) formula.

- In case that $\langle \emptyset \rangle \psi'$ is a CTL formula, following [Song and Touili, 2011], an AMA $\mathcal{M}_{\upsilon_A}$ with $\mathbf{O}(|\mathcal{P}||\psi'|)$ states and $|\Gamma|2^{\mathbf{O}(|\mathcal{P}||\psi'|)}$ transition rules that recognizes all configurations satisfying $\langle \emptyset \rangle \psi'$.
- In case that $\langle \emptyset \rangle \psi'$ is a CTL* formula, we construct a Büchi automaton B for ψ' and then reduce the problem to the emptiness problem of alternating Büchi pushdown systems. Following [Song and Touili, 2011], we can construct an AMA $\mathcal{M}_{\upsilon_A}$ of $|\mathcal{P}|2^{\mathbf{O}(|\psi'|)}$ states and $|\Gamma|2^{|\mathcal{P}|2^{\mathbf{O}(|\psi'|)}}$ transition rules that recognizes all configurations satisfying $\langle \emptyset \rangle \psi'$.

In either case, we have

Lemma 1. $\bigcup_{\upsilon_A} \mathcal{L}(\mathcal{M}_{\upsilon_A}) = \|\langle A \rangle \psi'\|^{\mathbf{ir}}_{\mathcal{P}}$.

It is not hard to see that one can construct an AMA $\mathcal{M}_{\phi'}$ such that $\mathcal{L}(\mathcal{M}_{\phi'}) = \bigcup_{\upsilon_A} \mathcal{L}(\mathcal{M}_{\upsilon_A})$. Because there are at most $|\mathsf{Ac}|^{|P||\Gamma||A|}$ collective **ir**-strategies for A and $|A| \leq |\mathsf{Ag}|$, we observe that $\mathcal{M}_{\phi'}$ contains at most $|\mathsf{Ac}|^{|P||\Gamma||\mathsf{Ag}|} \cdot \mathbf{O}(|\mathcal{P}||\psi'|)$ (resp. $|\mathsf{Ac}|^{|P||\Gamma||\mathsf{Ag}|} \cdot |\mathcal{P}|2^{\mathbf{O}(|\psi'|)}$) states and $|\mathsf{Ac}|^{|P||\Gamma||\mathsf{Ag}|} \cdot |\Gamma|2^{\mathbf{O}(|\mathcal{P}||\psi'|)}$ (resp. $|\mathsf{Ac}|^{|P||\Gamma||\mathsf{Ag}|} \cdot |\Gamma|2^{|\mathcal{P}|2^{\mathbf{O}(|\psi'|)}}$) transition rules if $\langle A \rangle \psi'$ is an ATEL$_{\mathbf{ir}}$ (resp. ATEL$^*_{\mathbf{ir}}$) formula. Since the membership problem of AMAs can be solved in polynomial time (cf. Proposition 1), we deduce that the model checking of ϕ' on $\mathcal{P}$ can be solved in EXPTIME for ATEL$_{\mathbf{ir}}$ and 2EXPTIME for ATEL$^*_{\mathbf{ir}}$.

ATEL$_{\mathbf{ir}}$/ATEL$^*_{\mathbf{ir}}$ formulae. Given an ATEL$_{\mathbf{ir}}$/ATEL$^*_{\mathbf{ir}}$ formula ϕ, we inductively compute an AMA $\mathcal{M}_{\phi'}$ from the state subformulae ϕ' such that $\mathcal{L}(\mathcal{M}_{\phi'}) = \|\phi'\|^{\mathbf{ir}}_{\mathcal{P}}$. The base case for atomic propositions is trivial. For the induction step:

- For ϕ' of the form $\neg q$, $\phi_1 \wedge \phi_2$ or $\phi_1 \vee \phi_2$, $\mathcal{M}_{\phi}$ can be computed by applying Boolean operations on $\mathcal{M}_{\phi_1}/\mathcal{M}_{\phi_2}$.
- For ϕ' of the form $\langle A \rangle \psi'$, we first compute a principal formula ϕ'' by replacing each state subformula ϕ''' in ψ' by a fresh atomic proposition $q_{\phi'''}$ and then saturate λ by setting $q_{\phi'''} \in \lambda(c)$ for every $c \in \mathcal{L}(\mathcal{M}_{\phi'''})$. By Lemma 1, we can construct an AMA $\mathcal{M}_{\phi''}$ from ϕ'' which is the desired AMA $\mathcal{M}_{\phi'}$. The formulae ϕ' of the form $[A]\psi'$ can be handled in a similar way.
- For ϕ' of the form $\mathbf{K}_i\phi''$ (resp. $\mathbf{E}_A\phi''$ and $\mathbf{C}_A\phi''$), suppose that the AMA $\mathcal{M}_{\phi''} = (S_1, \Gamma, \delta_1, I_1, S_f)$ recognizes $\|\phi''\|^{\mathbf{ir}}_{\mathcal{P}}$. Let $[p_1, \gamma_1], ..., [p_m, \gamma_m] \subseteq P \times \Gamma$ be the equivalence classes induced by the relation $\approx_i$ (resp. $\sim^E_A$ and $\sim^C_A$). We define the AMA $\mathcal{M}_{\phi'} = (P \cup \{s_f\}, \Gamma, \delta', P, \{s_f\})$, where δ' is defined as follows: for every $j \in [m]$, if $\{\langle p, \gamma\omega \rangle \mid (p, \gamma) \in [p_j, \gamma_j], \omega \in \Gamma^*\} \subseteq \mathcal{L}(\mathcal{M}_{\phi''})$, then for all $(p, \gamma) \in [p_j, \gamma_j]$ and $\gamma' \in \Gamma$, $\delta'(p, \gamma) = s_f$ and $\delta'(s_f, \gamma') = s_f$. The AMA $\mathcal{M}_{\phi'}$ for formulae ϕ' of the form $\overline{\mathbf{K}}_i\phi''$ (resp. $\overline{\mathbf{E}}_A\phi''$ and $\overline{\mathbf{C}}_A\phi''$) can be constructed similarly as for $\mathbf{K}_i\phi''$, using the condition $\{\langle p, \gamma\omega \rangle \mid (p, \gamma) \in [p_j, \gamma_j], \omega \in \Gamma^*\} \cap \mathcal{L}(\mathcal{M}_{\phi''}) \neq \emptyset$, instead of $\{\langle p, \gamma\omega \rangle \mid (p, \gamma) \in [p_j, \gamma_j], \omega \in \Gamma^*\} \subseteq \mathcal{L}(\mathcal{M}_{\phi''})$.

From the aforementioned arguments for principal formulae, we know that $\mathcal{M}_{\phi}$ contains at most $|\mathsf{Ac}|^{|P||\Gamma||\mathsf{Ag}|} \cdot \mathbf{O}(|\mathcal{P}||\phi|)$ (resp. $|\mathsf{Ac}|^{|P||\Gamma||\mathsf{Ag}|} \cdot |\mathcal{P}|2^{\mathbf{O}(|\phi|)}$) states and $|\mathsf{Ac}|^{|P||\Gamma||\mathsf{Ag}|} \cdot |\Gamma|2^{\mathbf{O}(|\mathcal{P}||\phi|)}$ (resp. $|\mathsf{Ac}|^{|P||\Gamma||\mathsf{Ag}|} \cdot |\Gamma|2^{|\mathcal{P}|2^{\mathbf{O}(|\phi|)}}$) transition rules if $\langle A \rangle \psi'$ is an ATEL$_{\mathbf{ir}}$ (resp. ATEL$^*_{\mathbf{ir}}$) formula. We then deduce from Theorem 4 and Proposition 1 the following result.

Theorem 5. *The model checking problems for ATEL$^*_{\mathbf{ir}}$ and ATEL$^*_{\mathbf{Ir}}$ (resp. ATEL$_{\mathbf{ir}}$ and ATEL$_{\mathbf{Ir}}$) on PEGSs with regular/simple valuations and regular/simple EARs are 2EXPTIME-complete (resp. EXPTIME-complete).*

We remark that in contrast to ATEL$^*_{\mathbf{IR}}$ (cf. Theorem 1), the complexity of ATEL$^*_{\mathbf{Ir}}$ (as well as ATEL$^*_{\mathbf{ir}}$) is lower as only imperfect recall is considered.

The lower bound follows from that of model checking problems for CTL/CTL* on PDSs with simple valuations [Walukiewicz, 2000; Bozzelli, 2007]. Namely, even for PEGSs with a single agent, perfect information, and simple valuations, the model checking problem is already EXPTIME/2EXPTIME-hard for CTL/CTL*.

6 Related Work

Model checking on finite-state CGSs under **IR** setting is well-studied in the literature [Alur *et al.*, 2002; Cermák *et al.*, 2015; Mogavero *et al.*, 2013; 2014b; 2014a]. The problem becomes undecidable for ATL on CGSs under **iR** setting [Dima and Tiplea, 2011]. Therefore, many works restrict to **ir** strategies [van der Hoek and Wooldridge, 2002; Schobbens, 2004; Jamroga and Dix, 2006; Jamroga, 2003; Bulling and Jamroga, 2011; Pilecki *et al.*, 2014; Cermák, 2015]. The model checking problem on PGSs under **IR** strategies were studied in [Murano and Perelli, 2015; Chen *et al.*, 2016a; 2016b], but only with perfect information. Furthermore, timed (resp. probabilistic) ATLs and timed (resp. probabilistic) CGSs were proposed to verify timed (resp. probabilistic) MASs, e.g., [Brihaye *et al.*, 2007; Chen *et al.*, 2013a]. These works are, however, orthogonal to the one reported in the current paper.

7 Conclusion and Future Work

In this paper, we have proposed PEGSs as a formal model of pushdown MASs with imperfect information. We showed that model checking is undecidable under size-preserving EARs, and provided optimal automata-theoretic algorithms under regular EARs.

Future work includes implementation of our algorithms and study of the model checking problem for PEGSs against SLK [Cermák, 2015].

References

[Alur *et al.*, 1997] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. In *FOCS'97*, pages 100–109, 1997.

[Alur *et al.*, 2002] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.

[Aminof *et al.*, 2013] Benjamin Aminof, Axel Legay, Aniello Murano, Olivier Serre, and Moshe Y. Vardi. Pushdown module checking with imperfect information. *Information and Computation*, 223:1–17, 2013.

[Ash, 2004] David W. Ash. Empowering automated trading in multi-agent environments. *Computational Intelligence*, 20(4):562–583, 2004.

[Bouajjani *et al.*, 1997] Ahmed Bouajjani, Javier Esparza, and Oded Maler. Reachability analysis of pushdown automata: Application to model-checking. In *CONCUR'97*, pages 135–150, 1997.

[Bozzelli, 2007] Laura Bozzelli. Complexity results on branching-time pushdown model checking. *Theoretical Computer Scicence*, 379(1-2):286–297, 2007.

[Brihaye *et al.*, 2007] Thomas Brihaye, François Laroussinie, Nicolas Markey, and Ghassan Oreiby. Timed concurrent game structures. In *CONCUR'07*, pages 445–459, 2007.

[Bulling and Jamroga, 2011] Nils Bulling and Wojciech Jamroga. Alternating epistemic mu-calculus. In *IJCAI'11*, pages 109–114, 2011.

[Cermák *et al.*, 2015] Petr Cermák, Alessio Lomuscio, and Aniello Murano. Verifying and synthesising multi-agent systems against one-goal strategy logic specifications. In *AAAI'15*, pages 2038–2044, 2015.

[Cermák, 2015] Petr Cermák. A model checker for strategy logic. Meng individual project, Department of Computing, Imperial College London, 2015.

[Chen *et al.*, 2013a] Taolue Chen, Vojtech Forejt, Marta Z. Kwiatkowska, David Parker, and Aistis Simaitis. Automatic verification of competitive stochastic systems. *Formal Methods in System Design*, 43(1):61–92, 2013.

[Chen *et al.*, 2016a] Taolue Chen, Fu Song, and Zhilin Wu. Global model checking on pushdown multi-agent systems. In *AAAI'16*, pages 2459–2465, 2016.

[Chen *et al.*, 2016b] Taolue Chen, Fu Song, and Zhilin Wu. Verifying pushdown multi-agent systems against strategy logics. In *IJCAI'16*, 2016.

[Clarke *et al.*, 2001] Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model checking*. MIT Press, 2001.

[Dima and Tiplea, 2011] Catalin Dima and Ferucio Laurentiu Tiplea. Model-checking ATL under imperfect information and perfect recall semantics is undecidable. *CoRR*, abs/1102.4225, 2011.

[Esparza *et al.*, 2003] Javier Esparza, Antonín Kucera, and Stefan Schwoon. Model checking LTL with regular valuations for pushdown systems. *Information and Computation*, 186(2):355–376, 2003.

[Fagin *et al.*, 1995] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning About Knowledge*. The MIT Press, 1995.

[Jamroga and Dix, 2006] Wojciech Jamroga and Jürgen Dix. Model checking abilities under incomplete information is indeed Delta2-complete. In *Proceedings of the 4th European Workshop on Multi-Agent Systems*, 2006.

[Jamroga, 2003] Wojciech Jamroga. Some remarks on alternating temporal epistemic logic. In *Proceedings of Formal Approaches to Multi-Agent Systems*, pages 133–140, 2003.

[Lomuscio and Raimondi, 2006] Alessio Lomuscio and Franco Raimondi. Model checking knowledge, strategies, and games in multi-agent systems. In *AAMAS'06*, pages 161–168, 2006.

[Mogavero *et al.*, 2013] Fabio Mogavero, Aniello Murano, and Luigi Sauro. On the boundary of behavioral strategies. In *LICS'13*, pages 263–272, 2013.

[Mogavero *et al.*, 2014a] Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe Y. Vardi. Reasoning about strategies: On the model-checking problem. *ACM Transations on Computational Logic*, 15(4):34:1–34:47, 2014.

[Mogavero *et al.*, 2014b] Fabio Mogavero, Aniello Murano, and Luigi Sauro. A behavioral hierarchy of strategy logic. In *CLIMA'14*, pages 148–165, 2014.

[Murano and Perelli, 2015] Aniello Murano and Giuseppe Perelli. Pushdown multi-agent system verification. In *IJCAI'15*, pages 1090–1097, 2015.

[Pilecki *et al.*, 2014] Jerzy Pilecki, Marek A. Bednarczyk, and Wojciech Jamroga. Model checking properties of multi-agent systems with imperfect information and imperfect recall. In *Proceedings of the 7th IEEE International Conference on Intelligent Systems*, pages 415–426, 2014.

[Schobbens, 2004] Pierre-Yves Schobbens. Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science*, 85(2):82–93, 2004.

[Song and Touili, 2011] Fu Song and Tayssir Touili. Efficient CTL model-checking for pushdown systems. In *CONCUR'11*, pages 434–449, 2011.

[Tsai *et al.*, 2005] Ming-Yu Tsai, Li-Chen Fu, and Ta-Chiun Chou. Automatic negotiation with mediated agents in e-commerce marketplace. In *EEE'05,*, pages 50–53. IEEE Computer Society, 2005.

[van der Hoek and Wooldridge, 2002] Wiebe van der Hoek and Michael Wooldridge. Tractable multiagent planning for epistemic goals. In *AAMAS'02*, pages 1167–1174, 2002.

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

[Walukiewicz, 2000] Igor Walukiewicz. Model checking CTL properties of pushdown systems. In *FSTTCS'00*, pages 127–138, 2000.

[Kupferman *et al.* , 2002] Orna Kupferman, Nir Piterman and Moshe Y. Vardi. Pushdown Specifications. In *LPAR'02*, pages 262–277, 2002.