Model Checking Pushdown Epistemic Game Structures*

Taolue Chen[1,4], Fu Song[2]✉, Zhilin Wu[3]

[1] Department of Computer Science and Information Systems, Birkbeck, University of London
[2] School of Information Science and Technology, ShanghaiTech University
[3] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences
[4] State Key Laboratory of Novel Software Technology, Nanjing University

Abstract. In this paper, we investigate the problem of verifying pushdown multi-agent systems with imperfect information. As the formal model, we introduce *pushdown epistemic game structures* (PEGSs), an extension of pushdown game structures with *epistemic accessibility relations* (EARs). For the specification, we consider extensions of alternating-time temporal logics with epistemic modalities: ATEL, ATEL* and AEMC. We study the model checking problems for ATEL, ATEL* and AEMC over PEGSs under various imperfect information settings. For ATEL and ATEL*, we show that size-preserving EARs, a common definition of the accessibility relation in the literature of games over pushdown systems with imperfect information, will render the model checking problem undecidable under imperfect information and imperfect recall setting. 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*. In contrast, for AEMC, we show that the model checking problem is EXPTIME-complete even in the presence of size-preserving EARs.

1 Introduction

Model checking, a well-studied method for automatic formal verification of complex systems, has been successfully applied to verify communication protocols, hardware designs and software, etc [15]. The key idea underlying the model checking method is to represent the system as a mathematical model, to express a desired property by a logic formula, and then to determine whether the formula is true in the model [15].

Recently, it has been extended to verify *multi-agent systems* (MASs), a novel paradigm which can be used to solve many complex tasks that might be difficult or inefficient for an individual agent to tackle. 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 [1,2]. Since then, a number of model checking algorithms for MASs have been studied for various models

* This work was partially supported by NSFC grant (61402179, 61532019, 61662035, 61572478, 61472474, 61100062, and 61272135), UK EPSRC grant (EP/P00430X/1), and European CHIST-ERA project SUCCESS.

and logics. For instance, [26] proposed a more expressive logic, called *strategy logic* (SL), which allows to express cooperation and enforcement of agents. However, the model checking problem for strategy logic on CGSs is NonElementarySpace-hard and the satisfiability problem is undecidable [23]. As a result, several fragments of strategy logic were investigated [10,24,25,23].

CGSs are usually determined by *Interpreted Systems* which are constructed via a Mealy-type or Moore-type synchronous composition of local transition systems of agents [18,22]. In the literature, local transition systems of each agent are usually *finite* (as, e.g., a finite 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 (e.g., pushdown scheduler [27]), or recourses shared between agents are unbounded [8], 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 (e.g., 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 [27,13,14]. In PGSs, agents do not posses their own local stacks, but can be seen as sharing a global stack. As the stack is unbounded, PGSs represent a class of *infinite-state* MASs, a proper extension of the finite-state MASs. PGSs allow, among others, modeling of unbounded memory or a unbound shared resource of agents, which is of particular importance in MASs [8,27].

On the logic side, one considers alternating-time temporal *epistemic* logics (ATEL, ATEL*) [33,19,29,20,28], alternating-time *epistemic* μ-calculus (AEMC) [7], and SLK [9], 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 [18]. 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, distributed systems, communication protocols, 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 [18,34].

This paper investigates model checking problems for ATEL, ATEL* and AEMC over *infinite-state* MASs under *imperfect* information setting. 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 in literature.

Model checking PEGSs depends crucially on how EARs are defined. A commonly adopted definition, called *size-preserving* EARs, was introduced for games over pushdown systems with imperfect information [3], where two configurations are deemed to be indistinguishable if the two stack contents are of the same size and, 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* over PEGSs are undecidable in general, even when restricted to imperfect recall (memoryless) 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. We first provide an automata-theoretic algorithm that solves the model checking problem for ATEL (resp. ATEL*) over PEGSs with simple EARs, then present a reduction from the model checking problems over PEGSs with regular EARs to the one over PEGSs with simple one. The algorithm runs in EXPTIME for ATEL and 2EXPTIME for ATEL*, and we show that these algorithms are optimal by giving matching lower bounds. In contrast, for AEMC, we show that the model checking problem is EXPTIME-complete, even in the presence of size-preserving EARs.

Related work. Model checking over finite-state CGSs under perfect information setting is well-studied in the literature [2,10,24,25,23]. The problem becomes undecidable for ATL on CGSs under imperfect information and perfect recall setting [16]. Therefore, many works restrict to imperfect information and imperfect recall strategies [33,29,20,19,7,28,9]. The model checking problem over PGSs under perfect information and perfect recall setting was studied in [27,13,14], 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., [6,11,12]. These works are, however, orthogonal to the one reported in the current paper.

Structure of the paper. In Section 2, we introduce pushdown epistemic game structures. In Section 3, we recall the definitions of ATEL, ATEL* and AEMC. In Section 4, we present the undecidable result for ATEL, ATEL* and propose model checking algorithms for decidable setting. The model checking algorithms for AEMC are presented in Section 5. Finally, we conclude in Section 6. Due to space restriction, all proofs are committed here which can be found in the accompanying technical report.

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); we assume that n is bounded;*
- 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 for all* $i \in [n]$, $\mathbf{d}(i) := a_i \in \mathsf{Ac}$*;*
- P *is a finite set of* control states*;*
- Γ *is a finite* stack alphabet*;*

– $\Delta : P \times \Gamma \times \mathcal{D} \rightarrow P \times \Gamma^*$ *is a transition function*[1]*;*
– $\lambda : P \times \Gamma^* \rightarrow 2^{\mathbf{AP}}$ *is a valuation that assigns to each configuration (i.e., an element of* $P \times \Gamma^*$*) a set of atomic propositions (i.e., observations);*
– $\sim_i \subseteq (P \times \Gamma^*) \times (P \times \Gamma^*)$ *is an* epistemic accessibility relation *(EAR) which is an equivalence relation.*

A *concurrent epistemic game structure* (CEGS) is a tuple $\mathcal{P} = (\mathsf{Ag}, \mathsf{Ac}, P, \Delta, \lambda, \{\sim_i \mid i \in \mathsf{Ag}\})$ where $\Delta : P \times \mathcal{D} \rightarrow P$, $\mathsf{Ag}, \mathsf{Ac}, P$ are defined similarly as PEGS, whereas 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 $i \in \mathsf{Ag}$. Hence, a PGS $\mathcal{P}$ is usually denoted as $(\mathsf{Ag}, \mathsf{Ac}, P, \Gamma, \Delta, \lambda)$.

A *configuration* of the PEGS $\mathcal{P}$ is a pair $\langle p, \omega \rangle$, where $p \in P$ and $\omega \in \Gamma^*$. 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 $\omega' \in \Gamma^*$, 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 onto 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, $\pi_{\leq i}$ denote $c_0 \ldots c_i$ and $\pi_{\geq i}$ denote $c_i c_{i+1} \ldots$. Given two tracks π and π', π and π' are *indistinguishable* for an agent $i \in \mathsf{Ag}$, denoted by $\pi \sim_i \pi'$, if $|\pi| = |\pi'|$ and 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 [29,7] classified by whether the action chosen by an agent relies on the whole history of past configurations or the current configuration, and whether the whole information is visible or not. Formally, the four types of strategies are defined as follows: where $\mathbf{i}$ (resp. $\mathbf{I}$) denotes imperfect (resp. perfect) information and $\mathbf{r}$ (resp. $\mathbf{R}$) denotes imperfect (resp. perfect) recall,

– **Ir** strategy is a function $\theta_i : C_{\mathcal{P}} \rightarrow \mathsf{Ac}$, i.e., the action made by the agent i depends on the current configuration;
– **IR** strategy is a function $\theta_i : \mathsf{Trks}_{\mathcal{P}} \rightarrow \mathsf{Ac}$, i.e., the action made by the agent i depends on the history, i.e. the sequence of configurations visited before;

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

- **ir** strategy is a function $\theta_i : \mathcal{C}_\mathcal{P} \to \mathsf{Ac}$ such that for all configurations $c, c' \in \mathcal{C}_\mathcal{P}$, 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.

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 [3], 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* is undecidable under this type of EARs, even with imperfect recall (cf. Theorem 3). To gain decidability, in this paper, we introduce *regular EARs* and a special case thereof, i.e. *simple* EARs. We remark that regular EARs align to the regular valuations (see later in this section) of atomic propositions, can been seen as approximations of size-preserving 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 all $\langle p, \gamma\omega \rangle, \langle p_1, \gamma_1\omega_1 \rangle \in \mathcal{C}_\mathcal{P}$, $\langle p, \gamma\omega \rangle \sim_i \langle p_1, \gamma_1\omega_1 \rangle$ 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 contents (excluding the tops), 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 the 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 all words $\omega, \omega_1 \in \Gamma^*$, $\Delta_i^*(s_{i,0}, \omega^R) = \Delta_i^*(s_{i,0}, \omega_1^R)$, that is, $\mathcal{A}_i$ contains only one state. Therefore, a simple EAR can be expressed by an equivalence relation $\approx_i$ on $P \times \Gamma$.

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 $\sim_A^E$. We use $|\mathcal{P}|$ to denote $|P| + |\Gamma| + |\Delta| + \prod_{i \in \mathsf{Ag}} |S_i|$.

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

Regular valuations. The model checking problem for pushdown systems (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 [21]. To gain decidability, we consider valuations specified by a function l which associates each pair $(p, q) \in P \times \mathbf{AP}$ with a DFA $\mathcal{A}_{p,q} = (S_{p,q}, \Gamma, \Delta_{p,q}, s_{p,q,0}, F_{p,q})$. This is usually referred to as a *regular valuation* [17]. The function l can be lifted to the valuation $\lambda_l : P \times \Gamma^* \to 2^{\mathbf{AP}}$: for every $\langle p, \omega\rangle \in C_{\mathcal{P}}$, $\lambda_l(\langle p, \omega\rangle) = \{q \in \mathbf{AP} \mid \Delta^*_{p,q}(\omega^R) \in F_{p,q}\}$.

A *simple valuation* is a regular valuation l such that for every $q \in \mathbf{AP}$, $p \in P, \gamma \in \Gamma$, and $\omega \in \Gamma^*$, it holds that $\Delta^*_{p,q}(\omega^R\gamma) = \Delta^*_{p,q}(\gamma)$, i.e., the truth of an atomic proposition only depends on the control state and the top of the stack. Let $|\lambda|$ denote the number of states of the product automaton of all the DFA's that represents λ.

Alternating Multi-Automata. In order 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. *[4] An AMA is a tuple $\mathcal{M} = (S, \Gamma, \delta, I, S_f)$, where S is a finite set of states, 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. An AMA is* multi-automaton *(MA) if for all $(s, S') \in \delta$, $|S'| \leq 1$.*

If $(s, \gamma, \{s_1, ..., s_m\}) \in \delta$, we will write $s \xrightarrow{\gamma} \{s_1, ..., s_m\}$ instead. We define the relation $\longrightarrow^* \subseteq S \times \Gamma^* \times 2^S$ as the least relation such that the following conditions hold:

- $s \xrightarrow{\epsilon}{}^* \{s\}$, for every $s \in S$;
- $s \xrightarrow{\gamma\omega}{}^* \bigcup_{i\in[m]} S_i$, if $s \xrightarrow{\gamma} \{s_1, ..., s_m\}$ and $s_i \xrightarrow{\omega}{}^* S_i$ 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 $p \xrightarrow{\omega}{}^* S'$. Let $\mathcal{L}(\mathcal{M})$ denote the set of all configurations accepted by $\mathcal{M}$.

Proposition 1. *[4] The membership problem of AMAs can be decided in polynomial time. AMAs are closed under all Boolean operations.*

Example 1. We illustrate our model by a modified example on the departmental travelling budget from [8]. Consider a department consisting of two professors 1, 2 and three lecturers 3, 4, and 5. The department's base budget (say 10 units) is allocated annually and can be spent to attend conferences or apply for grants, at most twice for each professor and at most once for each lecturer. Suppose there are two categories to request money to attend a conference: 1 unit or 2 units depending on whether it is *early* or *late* registration. Parts of a successful grant application will be credited to the department's budget. Suppose 3 units for each successful grant application will be added into the budget. A successful grant application from a member will immediately decrements 1 of his/her times using the budget. But, there is no a priori bound on the total budget, and no one can know how much or which way each of others has used budget. Therefore, all departmental members compete for the budget with imperfect information.

We can model this system as a PEGS $\mathcal{P}$ as follows. Each departmental member $i \in \{1, ..., 5\}$ is modeled as an agent which has actions {idle, AG, AC}, where AG denotes "applying for a grant", AC denotes "attending a conference", and idle denotes other actions without any cost. There is an additional agent 6 denoting the environment which decides whether the application is granted or not by nondeterministically choosing an action from $\{\mathsf{award}_i, \mathsf{reject}_i \mid i \in \{1, ..., 5\}\}$. Each control state of $\mathcal{P}$ is a tuple of local states of all the agents, where the local states of each agent encode the number of times that the agent has used the budget, namely, the local state $p_{i,k}$ denotes that the number is k for agent i. The submitted grant applications are recorded into the local states of the environment agent. The available units of budget are encoded into the stack, where the length of the stack content denotes the number of available units. Each decision made by the agents determines the stack operation according to the total costs of actions in the decision. Pushing m symbols onto the stack denotes that m units are added into the budget. Similarly, popping m symbols from the stack denotes that m units are consumed from the budget [3]. Therefore, the length of stack content restricts the chosen of actions by agents. This means that we only need one stack symbol for the stack alphabet. The transition rules of $\mathcal{P}$ can be constructed accordingly.

For this system, we can use size-preserving EARs to represent the constraint that each departmental member chooses the same action at two different scenaria but its local states and the number of available units are identical. In particular, for each agent $i \in \{1, ..., 5\}$ and two configurations c, c' of $\mathcal{P}$, $c \sim_i c'$ iff the local states of i in c, c', as well as lengths of stack contents in c, c', are the same.

On the other hand, it is also natural to assume that each departmental member chooses the same action at two different scenaria when its local states are identical, and the numbers of available units are either equal, or both greater than some bound (e.g., 6 units). This assumption can be described using regular EARs.

3 Specification Logics: ATEL, ATEL* and AEMC

In this section, we recall the definition of alternating-time temporal epistemic logics: ATEL [33], ATEL* [19] and AEMC [7], which were introduced for reasoning about knowledge and cooperation of agents in multi-agent systems. Informally, ATEL, ATLE* and AEMC can be considered as extensions of ATL, ATL* and AMC 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).

3.1 ATEL$_\sigma$ and ATEL$^*_\sigma$ (where $\sigma \in \{\mathrm{Ir}, \mathrm{IR}, \mathrm{ir}, \mathrm{iR}\}$)

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

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

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

[3] Since normal PEGS only pops one symbol from the stack at one step, in order to pop m symbols, we need to introduce some additional control states as done in [30].

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

The semantics of ATEL_σ^* 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_σ^* state formula, and $c \in C_\mathcal{P}$ be a configuration of $\mathcal{P}$. The satisfiability relation $\mathcal{P}, c \models_\sigma \phi$ is defined inductively on the structure of ϕ.

- $\mathcal{P}, c \models_\sigma q$ iff $q \in \lambda(c)$; $\;-\mathcal{P}, c \models_\sigma \neg q$ iff $q \notin \lambda(c)$;
- $\mathcal{P}, c \models_\sigma \phi_1 \vee \phi_2$ iff $\mathcal{P}, c \models_\sigma \phi_1$ or $\mathcal{P}, c \models_\sigma \phi_2$;
- $\mathcal{P}, c \models_\sigma \phi_1 \wedge \phi_2$ iff $\mathcal{P}, c \models_\sigma \phi_1$ and $\mathcal{P}, c \models_\sigma \phi_2$;
- $\mathcal{P}, c \models_\sigma \langle A \rangle \psi$ 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 $\pi \in \mathsf{out}^\sigma(c, \upsilon_A)$ such that $\mathcal{P}, \pi \models_\sigma \psi$;
- $\mathcal{P}, c \models_\sigma \mathbf{K}_i \phi$ 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 $\overline{\mathbf{K}}_i \phi$, but we use the relations $\sim_A^E$ 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., [15] for details. We denote by $\|\phi\|_\mathcal{P}^\sigma = \{c \in C_\mathcal{P} \mid \mathcal{P}, c \models_\sigma \phi\}$ the set of configurations satisfying ϕ. The *model checking problem* is to decide whether $c \in \|\phi\|_\mathcal{P}^\sigma$ for a given configuration c.

ATEL_σ is a syntactical fragment of ATEL_σ^* with restricted path formulae of the form

$$\psi ::= \mathbf{X}\ \phi \ \mid\ \mathbf{G}\ \phi \mid\ \phi\ \mathbf{U}\ \phi.$$

An ATEL_σ (resp. ATEL_σ^*) formula ϕ is *principal* if ϕ is in the form of $\langle A \rangle \psi$ or $[A]\psi$ such that ψ is a LTL formula. For instance, $\langle\{1\}\rangle \mathbf{F}\ q$ is a principal formula, while neither $\langle\{1\}\rangle \mathbf{F}(q \wedge \langle\{2\}\rangle \mathbf{G}\ q')$ nor $\langle\{1\}\rangle \mathbf{F}(\mathbf{K}_2\ q)$ is.

Example 2. Recall Example 1. Suppose that there are atomic propositions q_3, q_4, q_5 such that for each $i \in \{3,4,5\}$, $q_i \in \lambda(c)$ iff the configuration c contains the local state $p_{i,1}$, i.e., the agent i attends a conference. In addition, the atomic propositions g_i for $i \in \{1,2\}$ denote that agent i has applied for some grants. Consider the formula: $\phi_1 \triangleq \langle\{3,4,5\}\rangle \mathbf{F}(q_3 \wedge q_4 \wedge q_5)$, $\phi_2 \triangleq \langle\{2,3,4,5\}\rangle \mathbf{F}(q_3 \wedge q_4 \wedge q_5)$ and $\phi_3 \triangleq \mathbf{E}_{\{3,4,5\}} \langle\{3,4,5\}\rangle ((\mathbf{F}(g_1 \vee g_2)) \implies \mathbf{F}(q_3 \wedge q_4 \wedge q_5))$. ϕ_1 expresses that three lecturers have strategies such that all of them can attend some conferences. Obviously, ϕ_1 does not hold when both two professors attended conferences twice with late registrations, which costs 8 units. ϕ_2 expresses that three lecturers together with professor 2 have strategies such that all the lecturers can attend some conferences. ϕ_3 states that all three lecturers know that they have strategies such that if some professor applies for some grants, then all of them can attend some conferences. Obviously, ϕ_2 and ϕ_3 hold.

3.2 AEMC$_\sigma$ (where $\sigma \in \{\mathbf{Ir}, \mathbf{IR}, \mathbf{ir}, \mathbf{iR}\}$)

Definition 4 (Alternating-Time Epistemic μ-Calculus). *Given a finite set of propositional variables $\mathcal{Z}$, AEMC$_\sigma$ formulae are defined by the following grammar:*

$$\phi ::= q \mid \neg q \mid Z \mid \phi \vee \phi \mid \phi \wedge \phi \mid \langle A \rangle \mathbf{X}\phi \mid [A]\mathbf{X}\phi \mid \\ \mu Z.\phi \mid \nu Z.\phi \mid \mathbf{K}_i\phi \mid \mathbf{E}_A\phi \mid \mathbf{C}_A\phi \mid \overline{\mathbf{K}}_i\phi \mid \overline{\mathbf{E}}_A\phi \mid \overline{\mathbf{C}}_A\phi$$

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

The variables $Z \in \mathcal{Z}$ in the definition of AEMC$_\sigma$ are monadic second-order variables with the intention to represent a set of configurations of PEGSs. An occurrence of a variable $Z \in \mathcal{Z}$ is said to be *closed* in an AEMC$_\sigma$ formula ϕ if the occurrence of Z is in ϕ_1 for some subformula $\mu Z.\ \phi_1$ or $\nu Z.\ \phi_1$ of ϕ. Otherwise, the occurrence of Z in ϕ is said to be *free*. An AEMC$_\sigma$ formula ϕ is *closed* if it contains no free occurrences of variables from $\mathcal{Z}$.

The semantics of AEMC$_\sigma$ can be defined in an obvious way, where temporal modalities $\langle A \rangle \mathbf{X}\phi$ and $[A]\mathbf{X}\phi$ and epistemic modaliteis can be interpreted as in ATEL$^*_\sigma$ and the fixpoint modalities can be interpreted as in alternating mu-calculus [2]. Given a PEGS $\mathcal{P} = (\mathsf{Ag}, \mathsf{Ac}, P, \Gamma, \Delta, \lambda, \{\sim_i \mid i \in \mathsf{Ag}\})$, and a closed formula ϕ, the denotation function $\|\circ\|^\sigma_\mathcal{P}$ maps AEMC$_\sigma$ formulae to sets of configurations. A configuration c satisfies ϕ iff $c \in \|\phi\|^\sigma_\mathcal{P}$.

For closed AEMC$_\sigma$ formula ϕ, $\|\phi\|^\sigma_{\mathcal{P},\xi}$ is independent of ξ. Therefore, the superscript ξ will be dropped from $\|\phi\|^\sigma_{\mathcal{P},\xi}$, for closed AEMC$_\sigma$ formula ϕ. In addition, the subscript $\mathcal{P}$ is also dropped from $\|\phi\|^\sigma_{\mathcal{P},\xi}$ and $\|\phi\|^\sigma_\mathcal{P}$ when it is clear.

We remark that, for AEMC$_\sigma$ (where $\sigma \in \{\mathbf{Ir}, \mathbf{IR}, \mathbf{ir}, \mathbf{iR}\}$), it makes no difference whether the strategies are perfect recall or not, since each occurrence of the modalities $\langle A \rangle \mathbf{X}\phi$ and $[A]\mathbf{X}\phi$ will "reset" the strategies of agents. Therefore, we will ignore $\mathbf{R}$ and $\mathbf{r}$ and use AEMC$_\mathbf{I}$/AEMC$_\mathbf{i}$ to denote AEMC under perfect/imperfect information.

Proposition 2. *[7] For any closed AEMC$_\sigma$ formula ϕ and a PEGS $\mathcal{P}$, $\|\phi\|^{\mathbf{ir}}_\mathcal{P} = \|\phi\|^{\mathbf{iR}}_\mathcal{P}$ and $\|\phi\|^{\mathbf{Ir}}_\mathcal{P} = \|\phi\|^{\mathbf{IR}}_\mathcal{P}$.*

We mention that, although ATEL$_\mathbf{IR}$ and ATEL$^*_\mathbf{IR}$ can be translated into AEMC$_\mathbf{I}$, this is *not* the case for imperfect information. Namely, ATEL$_\mathbf{iR}$, ATEL$_\mathbf{ir}$, ATEL$^*_\mathbf{iR}$, and ATEL$^*_\mathbf{ir}$ cannot be translated into AEMC$_\mathbf{i}$. The interested readers are referred to [7] for more discussions.

CTL, CTL* and μ-calculus are special cases of ATL$_\sigma$, ATL$^*_\sigma$ and AMC in which all the modalities $\langle A \rangle \psi$ and $[A]\psi$ satisfy $A = \emptyset$ [4], while ATL$_\sigma$, ATL$^*_\sigma$ and AMC$_\sigma$ are special cases of ATEL$_\sigma$, ATEL$^*_\sigma$ and AEMC$_\sigma$ in which no epistemic modalities occur.

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

Theorem 1 ([13]). *The model checking problem for ATEL$_\mathbf{IR}$/AEMC$_\mathbf{IR}$ over PEGSs is EXPTIME-complete, and for ATEL$^*_\mathbf{IR}$ 3EXPTIME-complete .*

[4] $\langle\emptyset\rangle$ (resp. $[\emptyset]$) is the universal (resp. existential) path quantification A (resp. E).

Remark 1. In [7], 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 [7] 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 [7] 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. Similar differences exist for AEMC$_\sigma$. We decide to make the hidden epistemic modalities $\mathbf{E}_A$ explicit in this paper.

4 ATEL and ATEL* Model Checking

We first recall the following undecidability result.

Theorem 2 ([16]). *The model checking problem for ATL$_{\mathbf{iR}}$ and ATL$^*_{\mathbf{iR}}$ over CEGSs is undecidable.*

In light of Theorem 1 and Theorem 2, in this section, we focus on the model checking problems for ATEL$_{\mathbf{ir}}$/ATEL$^*_{\mathbf{ir}}$.

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 ATL$_{\mathbf{iR}}$ over CEGSs to the one for ATL$_{\mathbf{ir}}$ over PEGSs. From Theorem 2, we deduce the following result.

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

Theorem 3 rules out model checking algorithms for ATEL$_{\mathbf{ir}}$/ATEL$^*_{\mathbf{ir}}$ when the PEGS is equipped with size-preserving EARs. As mentioned before, we therefore consider the case with regular/simple EARs. We first consider the model checking problem over PEGSs with simple EARs. This will be solved by a reduction to the model checking problem for CTL/CTL* over pushdown systems [31,17]. We then provide a reduction from the model checking problem over PEGSs with regular EARs to the one over PEGSs with simple EARs. The main idea of the reduction, which is inspired by the reduction of PDSs with regular valuations to PDSs with simple valuations in [17], is to store the runs of DFAs representing the regular EARs into the stack.

4.1 Pushdown Systems

Definition 5. *A* pushdown system *(PDS) is a tuple $\mathcal{P} = (P, \Gamma, \Delta, \lambda)$, where P, Γ, λ are defined as for PEGSs, and $\Delta \subseteq (P\times\Gamma)\times(P\times\Gamma^*)$ is a finite set of transition rules.*

A configuration of $\mathcal{P}$ is an element $\langle p,\omega\rangle$ of $P\times\Gamma^*$. We write $\langle p,\gamma\rangle \hookrightarrow \langle q,\omega\rangle$ instead of $((p,\gamma),(q,\omega)) \in \Delta$. If $\langle p,\gamma\rangle \hookrightarrow \langle q,\omega\rangle$, then for every $\omega' \in \Gamma^*$, $\langle q,\omega\omega'\rangle$ is a *successor* of $\langle p,\gamma\omega'\rangle$. Given a configuration c, a path π of $\mathcal{P}$ starting from c is a sequence of configurations $c_0c_1...$ such that $c_0 = c$ and for all $i > 0$, c_i is a successor of c_{i-1}. Let $\prod_\mathcal{P}(c) \subseteq C^\omega_\mathcal{P}$ denote the set of all paths in $\mathcal{P}$ starting from c onwards.

Given a configuration c and a CTL/CTL* formula ϕ, the satisfiability relation $\mathcal{P}, c \models \phi$ is defined in a standard way (cf. [31,17]). For instance, $\mathcal{P}, c \models \langle\emptyset\rangle\psi$ iff $\forall\pi \in \prod_\mathcal{P}(c)$, $\mathcal{P},\pi \models \psi$, $\mathcal{P}, c \models [\emptyset]\psi$ iff $\exists\pi \in \prod_\mathcal{P}(c)$, $\mathcal{P},\pi \models \psi$. Let $\|\phi\|_\mathcal{P} = \{c \in C_\mathcal{P} \mid \mathcal{P}, c \models \phi\}$.

Theorem 4. *[17] Given a PDS $\mathcal{P} = (P, \Gamma, \Delta, \lambda)$ and a CTL/CTL* formula ϕ such that all state subformulae in ϕ are atomic propositions, we can effectively compute a MA $\mathcal{M}$ with $\mathbf{O}(|\lambda| \cdot |P| \cdot |\Delta| \cdot k)$ states in $\mathbf{O}(|\lambda| \cdot |P|^2 \cdot |\Delta| \cdot k)$ time such that the MA exactly recognizes $\|\phi\|_{\mathcal{P}}$, where k is $2^{\mathbf{O}(|\phi|)}$ (resp. $\mathbf{O}(|\phi|)$) for CTL* (resp. CTL). Moreover, a DFA $\mathcal{A} = (S, \Gamma, \Delta_1, s_0)$ with $\mathbf{O}(|\lambda| \cdot |\Delta| \cdot 2^{|P| \cdot k})$ states and a tuple of sets of accepting states $(F_p)_{p \in P}$ can be constructed in $\mathbf{O}(|\lambda| \cdot |\Delta| \cdot 2^{|P| \cdot k})$ time such that for every configuration $\langle p, \omega \rangle \in P \times \Gamma^*$, $\langle p, \omega \rangle \in \mathcal{L}(\mathcal{M})$ iff $\Delta_1^*(s_0, \omega^R) \in F_p$.*

4.2 Model Checking for PEGSs with Simple EARs

In this subsection, we propose an automatic-theoretic approach for solving the model checking problems for ATEL$_{\mathbf{ir}}$ and ATEL$^*_{\mathbf{ir}}$ over PEGSs with simple EARs.

Let us fix the ATEL$_{\mathbf{ir}}$/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 a regular valuation l represented by DFAs $(\mathcal{A}_{p,q})_{p \in P, q \in \mathbf{AP}}$ and $\sim_i$ is specified by an equivalence relation $\approx_i$ on $P \times \Gamma$ for $i \in \mathsf{Ag}$.

The idea of the algorithm is to construct, for each state subformula ϕ' of ϕ, an MA $\mathcal{M}_{\phi'}$ to represent the set of configurations satisfying ϕ'. We will first illustrate the construction in case that $\phi' = \langle A \rangle \psi$ (resp. $\phi' = [A]\psi$) is a principal formula, then extend the construction to the more general case.

Principal Formulae. Our approach will reduce the model checking problem over PEGSs to the model checking problem for CTL/CTL* over 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* $\approx_i$, namely, 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 $\mathbf{ir}$-strategy υ_i of i. Therefore, any $\mathbf{ir}$-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., $\upsilon_i : P \times \Gamma \to \mathsf{Ac}$ such that $\upsilon_i(p, \gamma) = \upsilon_i(p', \gamma')$ for all (p, γ) and (p', γ') with $(p, \gamma) \approx_i (p', \gamma')$.

Proposition 3. *Given a configuration $c \in C_{\mathcal{P}}$ and a set of agents $A \subseteq \mathsf{Ag}$, the following statements hold:*

i. for any collective $\mathbf{ir}$-strategy υ_A such that $\upsilon_A(i)$ respects to $\approx_i$ for $i \in A$, there exist functions $\upsilon'_i : P \times \Gamma \to \mathsf{Ac}$ for $i \in A$ such that $\mathsf{out}^{\mathbf{ir}}(c, \upsilon_A) = \mathsf{out}(c, \bigcup_{i \in A} \upsilon'_i)$ and $\upsilon'_i(p, \gamma) = \upsilon'_i(p', \gamma')$ for all (p, γ) and (p', γ') with $(p, \gamma) \approx_i (p', \gamma')$;
ii. for any function $\upsilon'_i : P \times \Gamma \to \mathsf{Ac}$ for $i \in A$ such that $\upsilon'_i(p, \gamma) = \upsilon'_i(p', \gamma')$ for all (p, γ) and (p', γ') with $(p, \gamma) \approx_i (p', \gamma')$, there exists a collective $\mathbf{ir}$-strategy υ_A such that $\upsilon_A(i)$ respects to $\approx_i$ for $i \in A$ and $\mathsf{out}^{\mathbf{ir}}(c, \upsilon_A) = \mathsf{out}(c, \bigcup_{i \in A} \upsilon'_i)$;

where $\mathsf{out}(c, \bigcup_{i \in A} \upsilon'_i)$ denotes the set of all paths $\pi = \langle p_0, \gamma_0\omega_0 \rangle \langle p_1, \gamma_1\omega_1 \rangle \cdots$ such that $\langle p_0, \gamma_0\omega_0 \rangle = c$ and for all $k \geq 0$, there exists $\mathbf{d}_k \in \mathcal{D}$ such that $\langle p_k, \gamma_k\omega_k \rangle \overset{\mathbf{d}_k}{\Longrightarrow}_{\mathcal{P}} \langle p_{k+1}, \gamma_{k+1}\omega_{k+1} \rangle$ and $\mathbf{d}_k(i) = \upsilon'_i(p_k, \gamma_k)$ for all $i \in A$.

According to Proposition 3, we can check all the possible collective $\mathbf{ir}$-strategies, as the number of possible functions from $P \times \Gamma \to \mathsf{Ac}$ is finite. Let us now fix a specific collective $\mathbf{ir}$-strategy $\upsilon_A = (\upsilon_i)_{i \in A}$ for A. For each $(p, \gamma) \in P \times \Gamma$, after applying a collective $\mathbf{ir}$-strategy $\upsilon_A = (\upsilon_i)_{i \in A}$ for A, we define a PDS $\mathcal{P}_{\upsilon_A} = (P, \Gamma, \Delta', \lambda)$, where Δ' is defined as follows: for every $p, p' \in P$, $\gamma \in \Gamma$ and $\omega \in \Gamma^*$,

$$((p, \gamma), (p', \omega)) \in \Delta' \text{ iff } \exists \mathbf{d} \in \mathcal{D} \; s.t. \; \forall i \in A,\; \mathbf{d}(i) = \upsilon_i(p, \gamma) \text{ and } \Delta(p, \gamma, \mathbf{d}) = (p', \omega).$$

Lemma 1. $\mathit{out}^{\mathbf{ir}}(c, \upsilon_A) = \prod_{\mathcal{P}_{\upsilon_A}}(c)$.

Following from Lemma 1, for $\phi' = \langle A\rangle\psi$, $\mathcal{P}, c \models_{\mathbf{ir}} \phi'$ iff there exists a collective **ir**-strategy υ_A such that for all paths $\pi \in \prod_{\mathcal{P}_{\upsilon_A}}(c)$, $\mathcal{P}, \pi \models_{\mathbf{ir}} \psi$. The latter holds iff there exists a collective **ir**-strategy υ_A such that $\mathcal{P}_{\upsilon_A}, c \models \langle\emptyset\rangle\psi$. Similarly, for $\phi' = [A]\psi$, $\mathcal{P}, c \models_{\mathbf{ir}} \phi'$ iff for all collective **ir**-strategies υ_A, there exists a path $\pi \in \prod_{\mathcal{P}_{\upsilon_A}}(c)$ such that $\mathcal{P}, \pi \models_{\mathbf{ir}} \psi$. The latter holds iff for all collective **ir**-strategies υ_A, $\mathcal{P}_{\upsilon_A}, c \models [\emptyset]\psi$.

Fix a collective **ir**-strategy υ_A with respect to $\approx_i$ for $i \in A$, by applying Theorem 4, we can construct a MA $\mathcal{M}_{\upsilon_A}$ such that $\mathcal{L}(\mathcal{M}_{\upsilon_A}) = \{c \in P \times \Gamma^* \mid \mathcal{P}_{\upsilon_A}, c \models \langle\emptyset\rangle\psi'\}$ (resp. $\mathcal{L}(\mathcal{M}_{\upsilon_A}) = \{c \in P \times \Gamma^* \mid \mathcal{P}_{\upsilon_A}, c \models [\emptyset]\psi'\}$). Since, there are at most $|\mathsf{Ac}|^{|P|\cdot|\Gamma|\cdot|A|}$ collective **ir**-strategies with respect to $\approx_i$ for $i \in A$ and $|A| \leq |\mathsf{Ag}|$, we can construct a MA $\mathcal{M}_{\phi'}$ such that $\mathcal{L}(\mathcal{M}_{\phi'}) = \bigcup_{\upsilon_A} \mathcal{L}(\mathcal{M}_{\upsilon_A})$ (resp. $\mathcal{L}(\mathcal{M}_{\phi'}) = \bigcap_{\upsilon_A} \mathcal{L}(\mathcal{M}_{\upsilon_A})$).

Lemma 2. *For every principal ATEL$^*_{\mathbf{ir}}$ (resp. ATEL$_{\mathbf{ir}}$) formula ϕ', we can construct a MA $\mathcal{M}_{\phi'}$ with $\mathbf{O}(|\mathsf{Ac}|^{|P|\cdot|\Gamma|\cdot|\mathsf{Ag}|} \cdot |\lambda| \cdot |P| \cdot |\Delta| \cdot k)$ states in $\mathbf{O}(|\mathsf{Ac}|^{|P|\cdot|\Gamma|\cdot|\mathsf{Ag}|} \cdot |\lambda| \cdot |P|^2 \cdot |\Delta| \cdot k)$ time such that the MA exactly recognizes $\|\phi'\|^{\mathbf{ir}}_{\mathcal{P}}$, where k is $2^{\mathbf{O}(|\phi|)}$ (resp. $\mathbf{O}(|\phi|)$). Moreover, a DFA $\mathcal{A} = (S, \Gamma, \Delta_1, s_0)$ with $\mathbf{O}(|\mathsf{Ac}|^{|P|\cdot|\Gamma|\cdot|\mathsf{Ag}|} \cdot |\lambda| \cdot |\Delta| \cdot 2^{|P|\cdot k})$ states and a tuple of sets of accepting states $(F_p)_{p\in P}$ can be constructed in $\mathbf{O}(|\mathsf{Ac}|^{|P|\cdot|\Gamma|\cdot|\mathsf{Ag}|} \cdot |\lambda| \cdot |\Delta| \cdot 2^{|P|\cdot k})$ time such that for every configuration $\langle p, \omega\rangle \in P \times \Gamma^*$, $\langle p, \omega\rangle \in \mathcal{L}(\mathcal{M}_{\phi'})$ iff $\Delta_1^*(s_0, \omega^R) \in F_p$.*

General ATEL$_{\mathbf{ir}}$/ATEL$^*_{\mathbf{ir}}$ formulae. We now present a model checking algorithm for general ATEL$_{\mathbf{ir}}$/ATEL$^*_{\mathbf{ir}}$ formulae. Given an ATEL$_{\mathbf{ir}}$/ATEL$^*_{\mathbf{ir}}$ formula ϕ, we inductively compute a MA $\mathcal{M}_{\phi'}$ from the state subformula 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 compute a new regular valuation λ' by saturating λ which sets $q_{\phi'''} \in \lambda(c)$ for $c \in \mathcal{L}(\mathcal{M}_{\phi'''})$. To saturate λ, we use the DFA transformed from $\mathcal{M}_{\phi'''}$. Similar to the construction in [17], $|\lambda'| = |\lambda| \cdot |\mathsf{Ac}|^{|P|\cdot|\Gamma|\cdot|\mathsf{Ag}|} \cdot 2^{|P|\cdot k}$, where k is $2^{\mathbf{O}(|\phi|)}$ (resp. $\mathbf{O}(|\phi|)$) for ATEL$^*_{\mathbf{ir}}$ (resp. ATEL$_{\mathbf{ir}}$). By Lemma 2, we can construct a MA $\mathcal{M}_{\phi''}$ from ϕ'' which is the desired MA $\mathcal{M}_{\phi'}$. The construction for $\mathcal{M}_{[A]\psi'}$ is similar.
- For ϕ' of the form $\mathbf{K}_i\phi''$ (resp. $\mathbf{E}_A\phi''$ and $\mathbf{C}_A\phi''$), suppose that the MA $\mathcal{M}_{\phi''} = (S_1, \Gamma, \delta_1, I_1, S_f)$ 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 MA $\mathcal{M}_{\phi'} = (P \cup \{s_f\}, \Gamma, \delta', P, \{s_f\})$, where 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 MA $\mathcal{M}_{\phi'}$ 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 $\{\langle p, \gamma\omega\rangle \mid (p, \gamma) \in [p_j, \gamma_j], \omega \in \Gamma^*\} \subseteq \mathcal{L}(\mathcal{M}_{\phi''})$.

In the above algorithm, MAs are transformed into DFAs at most $|\phi|$ times. Each transformation only introduces the factor $|\mathsf{Ac}|^{|P|\cdot|\Gamma|\cdot|\mathsf{Ag}|} \cdot 2^{|P|\cdot k}$ into $|\lambda|$ [17]. We then deduce the following result from Proposition 1 and Lemma 2.

Theorem 5. *The model checking problem for ATEL$^*_{\mathbf{ir}}$ over PEGSs with simple EARs is 2EXPTIME-complete, while the problem for ATEL$_{\mathbf{ir}}$ is EXPTIME-complete.*

Proof. The lower bound of the model checking problem for ATEL$^*_{\mathbf{ir}}$ follows from that the model checking problem for CTL* over PDSs with simple valuations [5] is 2EXPTIME-complete. Namely, even for PEGSs with a single agent, and simple valuations, the model checking problem is already 2EXPTIME-hard. The hardness for ATEL$_{\mathbf{ir}}$ follows from the fact that the model checking problem for CTL over PDSs is EXPTIME-complete[35,32]. □

4.3 Model Checking for PEGSs with Regular EARs

In this subsection, we present a reduction from the model checking problem over PEGSs with regular EARs to the problem over PEGSs with 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 $(\approx_i, \mathcal{A}_i)$, 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 $\boldsymbol{\mathcal{A}} = (\boldsymbol{S}, \Gamma, \boldsymbol{\delta}, \boldsymbol{s_0})$ be the product automaton of $\mathcal{A}_i$'s for $i \in \mathsf{Ag}$, such that $\boldsymbol{S} = S_1 \times ... \times S_n$, $\boldsymbol{s_0} = [s_{1,0}, ..., s_{n,0}]$, and $\boldsymbol{\delta}(\boldsymbol{s_1}, \gamma) = \boldsymbol{s_2}$ if for every $i \in [n]$, $\delta_i(s_{i,1}, \gamma) = s_{i,2}$, where $s_{i,j}$ denotes the state of $\mathcal{A}_i$ in $\boldsymbol{s_j}$.

We will construct a new PEGS $\mathcal{P}'$ with simple EARs such that the model checking problem over $\mathcal{P}$ is reduced to the problem over $\mathcal{P}'$. Intuitively, the PEGS $\mathcal{P}'$ with simple EARs to be constructed stores the state obtained by running $\boldsymbol{\mathcal{A}}$ over the reverse of the partial stack content up to the current position (exclusive) into the stack. Formally, the PEGS $\mathcal{P}'$ is given by $(\mathsf{Ag}, \mathsf{Ac}, P, \Gamma', \Delta', \lambda', \{\sim'_i \mid i \in \mathsf{Ag}\})$, where

- $\Gamma' = \Gamma \times \boldsymbol{S}$;
- for each $i \in \mathsf{Ag}$, $\sim'_i$ is specified by an equivalence relation $\approx'_i$ on $P \times \Gamma'$ defined as follows: $(p, [\gamma, \boldsymbol{s}]) \approx'_i (p', [\gamma', \boldsymbol{s'}])$ iff $(p, \gamma) \approx_i (p', \gamma')$ and $\boldsymbol{s} = \boldsymbol{s'}$;
- Δ' is defined as follows: for every state $\boldsymbol{s} \in \boldsymbol{S}$,
  1. for every $\langle p, \gamma\rangle \overset{\mathbf{d}}{\hookrightarrow}_{\mathcal{P}} \langle p', \epsilon\rangle$, $\langle p, [\gamma, \boldsymbol{s}]\rangle \overset{\mathbf{d}}{\hookrightarrow}_{\mathcal{P}'} \langle p', \epsilon\rangle$,
  2. for every $\langle p, \gamma\rangle \overset{\mathbf{d}}{\hookrightarrow}_{\mathcal{P}} \langle p', \gamma_k ... \gamma_1\rangle$ with $k \geq 1$ and $\boldsymbol{\delta}(\boldsymbol{s}_j, \gamma_j) = \boldsymbol{s}_{j+1}$ for every $j:\ 1 \leq j \leq k-1$ (where $\boldsymbol{s}_1 = \boldsymbol{s}$), then $\langle p, [\gamma, \boldsymbol{s}]\rangle \overset{\mathbf{d}}{\hookrightarrow}_{\mathcal{P}'} \langle p', [\gamma_k, \boldsymbol{s}_k] ... [\gamma_1, \boldsymbol{s}_1]\rangle$.

Finally, the valuation λ' is adjusted accordingly to λ, i.e., for every $\langle p', [\gamma_k, \boldsymbol{s}_k]...[\gamma_0, \boldsymbol{s}_0]\rangle \in C_{\mathcal{P}'}$, $\lambda'(\langle p', [\gamma_k, \boldsymbol{s}_k]...[\gamma_0, \boldsymbol{s}_0]\rangle) = \lambda(\langle p', \gamma_k...\gamma_0\rangle)$.

Lemma 3. *The model checking problem for ATEL$_{\mathbf{ir}}$ (resp. ATEL$^*_{\mathbf{ir}}$) over 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 problem over 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 $\Gamma \times \boldsymbol{S}$, where $\boldsymbol{S}$ is the state space of the product of $\mathcal{A}_i$'s for $i \in \mathsf{Ag}$.*

Theorem 6. *The model checking problem for ATEL$^*_{\mathbf{ir}}$ (resp. ATEL$_{\mathbf{ir}}$) over PEGSs with regular EARs is 2EXPTIME-complete (resp. EXPTIME-complete).*

5 AEMC Model Checking

In this section, we propose algorithms for the model checking problems for $\mathrm{AEMC}_{\mathbf{i}}$ over PEGSs with size-preserving/regular/simple EARs. At first, we remark that Theorem 3 does *not* hold for $\mathrm{AEMC}_{\mathbf{i}}$ (recall that $\mathrm{AEMC}_{\mathbf{i}} = \mathrm{AEMC}_{\mathbf{ir}} = \mathrm{AEMC}_{\mathbf{iR}}$). Indeed, we will show that the model checking problems for $\mathrm{AEMC}_{\mathbf{i}}$ over PEGSs with size-preserving/regular/simple EARs are EXMPTIME-complete.

Fix a closed $\mathrm{AEMC}_{\mathbf{i}}$ formula ϕ and a PEGS $\mathcal{P} = (\mathsf{Ag}, \mathsf{Ac}, P, \Gamma, \Delta, \lambda, \{\sim_i \mid i \in \mathsf{Ag}\})$ with size-preserving/regular/simple EARs. We will construct an AMA $\mathcal{A}_\phi$ to capture $||\phi||^{\mathbf{i}}_{\mathcal{P}}$ by induction on the syntax of $\mathrm{AEMC}_{\mathbf{i}}$ formulae.

Atomic formulae, Boolean operators, formulae of the form $\langle A\rangle \mathbf{X}\phi'$ and $[A]\mathbf{X}\phi'$, and fixpoint operators can be handled as in [13], where the model checking problem for AMC over PGSs was considered, as imperfect information does not play a role for these operators. In the sequel, we illustrate how to deal with the epistemic modalities. Regular/simple EARs can be tackled in a very similar way to Section 4, we focus on the size-preserving one.

Suppose size-preserving EARs $\sim_i$ for $i \in \mathsf{Ag}$ are specified by equivalence relations $\simeq_i \subseteq (P \times P) \cup (\Gamma \times \Gamma)$. For the formula $\phi = \mathbf{K}_i\phi'$, suppose the AMA $\mathcal{A}_{\phi'} = (S', \Gamma, \delta', I', S'_f)$ recognizing $||\phi'||^{\mathbf{ir}}_{\mathcal{P}}$ has been constructed. We construct $\mathcal{A}_\phi = (S', \Gamma, \delta, I, S'_f)$ as follows.

- $I = \{p \in P \mid \exists p' \in I'.\ p \simeq_i p'\}$.
- For each $(p, \gamma) \in P \times \Gamma$, let $[p]_{\simeq_i}$ (resp. $[\gamma]_{\simeq_i}$) be the equivalence of p (resp. γ) under $\simeq_i$, and $\overline{S'_{p,\gamma}} := \{S'_{p,\gamma} \mid (p, \gamma, S'_{p,\gamma}) \in \delta'\}$. Then $(p, \gamma, S) \in \delta$ if (1) for all $p' \in [p]_{\simeq_i}$ and $\gamma' \in [\gamma]_{\simeq_i}$, $\overline{S'_{p',\gamma'}} \neq \emptyset$; and (2) $S = \bigcup_{p' \in [p]_{\simeq_i}, \gamma' \in [\gamma]_{\simeq_i}} S''_{p',\gamma'}$, where $S''_{p',\gamma'} \in \overline{S'_{p',\gamma'}}$.
- For every $(s, \gamma, S) \in \delta'$ such that $s \in S' \setminus P$, let $(s, \gamma', S) \in \delta$ for every $\gamma' \in \Gamma$ with $\gamma' \simeq_i \gamma$.

For the formula $\phi = \overline{\mathbf{K}}_i\phi'$, suppose the AMA $\mathcal{A}_{\phi'} = (S', \Gamma, \delta', I', S'_f)$ recognizes $||\phi'||^{\mathbf{ir}}_{\mathcal{P}}$. We construct $\mathcal{A}_\phi = (S', \Gamma, \delta, I, S'_f)$ as follows.

- $I = \{p \in P \mid \exists p' \in I'.\ p \sim_i p'\}$.
- For each $(p, \gamma) \in P \times \Gamma$, if there is $(p', \gamma', S'_1) \in \delta'$ such that $p \simeq_i p'$ and $\gamma \simeq_i \gamma'$, let $(p, \gamma, S'_1) \in \delta$.
- For every $(s, \gamma, S) \in \delta'$ such that $s \in S' \setminus P$, let $(s, \gamma', S) \in \delta$ for every $\gamma' \in \Gamma$ with $\gamma' \simeq_i \gamma$.

The AMA $\mathcal{A}_\phi$ for ϕ of the form $\mathbf{E}_A\phi'$, $\mathbf{C}_A\phi'$, $\overline{\mathbf{E}}_A\phi'$ or $\overline{\mathbf{C}}_A\phi'$ can be constructed in a very similar way, in which the relation $\simeq_i$ is replaced by the relation $\bigcup_{i\in A} \simeq_i$ (resp. the transitive closure of $\bigcup_{i\in A} \simeq_i$).

Lemma 4. *Given a PEGS $\mathcal{P}$ with regular valuations and size-preserving EARs , and a closed $\mathrm{AEMC}_{\mathbf{i}}$ formula ϕ, we can construct an AMA $\mathcal{A}_\phi$ recognizing $||\phi||^{\mathbf{i}}_{\mathcal{P}}$ in exponential time with respect to $|\mathcal{P}|$, $|\lambda|$ and $|\phi|$.*

From Lemma 4 and Proposition 1, we have:

Theorem 7. *The model checking problem for $\mathrm{AEMC}_{\mathbf{i}}$ over PEGSs with regular/simple valuations and size-preserving/regular/simple EARs is EXPTIME-complete.*

The lower bound follows from fact that the model checking problem for AMC over PGSs with simple valuations is EXPTIME-complete [13].

6 Conclusion and Future Work

In this paper, we have shown that the model checking problem is undecidable for $\text{ATL}_{\mathbf{ir}}/\text{ATL}^*_{\mathbf{ir}}$ over PEGSs with size-preserving EARs, and provided optimal automata-theoretic model checking algorithms for $\text{ATEL}_{\mathbf{ir}}/\text{ATEL}^*_{\mathbf{ir}}$ over PEGSs with regular/simple EARs. We also have provided optimal model checking algorithms for $\text{AEMC}_{\mathbf{i}}$ over PEGSs under size-preserving/regular/simple EARs with matching lower bounds.

The model checking problem for $\text{ATEL}_{\mathbf{Ir}}/\text{ATEL}^*_{\mathbf{Ir}}$ or $\text{ATL}_{\mathbf{Ir}}/\text{ATL}^*_{\mathbf{Ir}}$ over PEGSs is still open. We note that the problem for $\text{ATEL}_{\mathbf{Ir}}/\text{ATEL}^*_{\mathbf{Ir}}$ or $\text{ATL}_{\mathbf{Ir}}/\text{ATL}^*_{\mathbf{Ir}}$ over CEGSs can be solved by nondeterministically choosing a strategy via selecting a subset of the transition relation, as the strategies only depend on control states yielding a finite set of possible strategies [29]. However, similar techniques are no longer applicable in PEGSs, as the strategies depend on stack contents apart from control states, which may yield an infinite set of possible strategies.

References

1. Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. In *FOCS'97*, pp. 100–109, 1997.
2. Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.
3. 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.
4. Ahmed Bouajjani, Javier Esparza, and Oded Maler. Reachability analysis of pushdown automata: Application to model-checking. In *CONCUR'97*, pp. 135–150, 1997.
5. Laura Bozzelli. Complexity results on branching-time pushdown model checking. *Theoretical Computer Scicence*, 379(1-2):286–297, 2007.
6. Thomas Brihaye, François Laroussinie, Nicolas Markey, and Ghassan Oreiby. Timed concurrent game structures. In *CONCUR'07*, pp. 445–459, 2007.
7. Nils Bulling and Wojciech Jamroga. Alternating epistemic mu-calculus. In *IJCAI'11*, pp. 109–114, 2011.
8. Nils Bulling and Hoang Nga Nguyen. Model checking resource bounded systems with shared resources via alternating Büchi pushdown systems. In *PRIMA'15*, pp. 640–649, 2015.
9. Petr Cermák. A model checker for strategy logic. Meng individual project, Department of Computing, Imperial College London, 2015.
10. Petr Cermák, Alessio Lomuscio, and Aniello Murano. Verifying and synthesising multi-agent systems against one-goal strategy logic specifications. In *AAAI'15*, pp. 2038–2044, 2015.
11. 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.
12. Taolue Chen, Vojtech Forejt, Marta Z. Kwiatkowska, David Parker, and Aistis Simaitis. Prism-games: A model checker for stochastic multi-player games. In *TACAS*, pp. 185–191, 2013.

13. Taolue Chen, Fu Song, and Zhilin Wu. Global model checking on pushdown multi-agent systems. In *AAAI'16*, pp. 2459–2465, 2016.
14. Taolue Chen, Fu Song, and Zhilin Wu. Verifying pushdown multi-agent systems against strategy logics. In *IJCAI'16*, pp. 180–186,2016.
15. Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model checking*. MIT Press, 2001.
16. Catalin Dima and Ferucio Laurentiu Tiplea. Model-checking ATL under imperfect information and perfect recall semantics is undecidable. *CoRR*, abs/1102.4225, 2011.
17. Javier Esparza, Antonín Kucera, and Stefan Schwoon. Model checking LTL with regular valuations for pushdown systems. *Information and Computation*, 186(2):355–376, 2003.
18. Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning About Knowledge*. The MIT Press, 1995.
19. Wojciech Jamroga. Some remarks on alternating temporal epistemic logic. In *FAMAS'03*, pp. 133–140, 2003.
20. Wojciech Jamroga and Jürgen Dix. Model checking abilities under incomplete information is indeed Delta2-complete. In *EUMAS'06*, 2006.
21. Orna Kupferman, Nir Piterman, and Moshe Y. Vardi. Pushdown specifications. In *LPAR'02*, pp. 262–277, 2002.
22. Alessio Lomuscio and Franco Raimondi. Model checking knowledge, strategies, and games in multi-agent systems. In *AAMAS'06*, pp. 161–168, 2006.
23. 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.
24. Fabio Mogavero, Aniello Murano, and Luigi Sauro. On the boundary of behavioral strategies. In *LICS'13*, pp. 263–272, 2013.
25. Fabio Mogavero, Aniello Murano, and Luigi Sauro. A behavioral hierarchy of strategy logic. In *CLIMA'14*, pp. 148–165, 2014.
26. Fabio Mogavero, Aniello Murano, and Moshe Y. Vardi. Reasoning about strategies. In *FSTTCS'10*, pages 133–144, 2010. pp.
27. Aniello Murano and Giuseppe Perelli. Pushdown multi-agent system verification. In *IJCAI'15*, pp. 1090–1097, 2015.
28. Jerzy Pilecki, Marek A. Bednarczyk, and Wojciech Jamroga. Model checking properties of multi-agent systems with imperfect information and imperfect recall. In *IS'14*, pp. 415–426, 2014.
29. Pierre-Yves Schobbens. Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science*, 85(2):82–93, 2004.
30. Stefan Schwoon. Model checking pushdown systems. PhD thesis, Technical University Munich, Germany, 2002.
31. Fu Song and Tayssir Touili. Efficient CTL model-checking for pushdown systems. In *CONCUR'11*, pp. 434–449, 2011.
32. Fu Song and Tayssir Touili. Efficient CTL model-checking for pushdown systems. *Theoretical Computer Scicence*, 549:127–145, 2014.
33. Wiebe van der Hoek and Michael Wooldridge. Tractable multiagent planning for epistemic goals. In *AAMAS'02*, pp. 1167–1174, 2002.
34. 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.
35. Igor Walukiewicz. Model checking CTL properties of pushdown systems. In *FSTTCS'00*, pp. 127–138, 2000.
36. Matthew Hague and C.-H. Luke Ong. A saturation method for the modal μ-calculus over pushdown systems. Information and Computation, 209(5):799–821, 2011.

A Appendix

A.1 Semantics of AEMC$_\sigma$

The semantics of AEMC$_\sigma$ is defined over PEGSs. A *valuation* $\xi : \mathcal{Z} \to 2^{C_\mathcal{P}}$ is a function assigning to each propositional variable a set of configurations. We use $\xi[Z \mapsto C]$ to denote the valuation which is equal to ξ except for $\xi[Z \mapsto C](Z) = C$. Given a PEGS $\mathcal{P} = (\mathsf{Ag}, \mathsf{Ac}, P, \Gamma, \Delta, \lambda, \{\sim_i \mid i \in \mathsf{Ag}\})$ and a valuation $\xi : \mathcal{Z} \to 2^{C_\mathcal{P}}$, the denotation function $\|\circ\|^\sigma_{\mathcal{P},\xi}$ that maps AEMC$_\sigma$ formulae to sets of configurations is inductively defined as follows:

- $\|q\|^\sigma_{\mathcal{P},\xi} = \{c \mid q \in \lambda(c)\}$;
- $\|\neg q\|^\sigma_{\mathcal{P},\xi} = C_\mathcal{P} \setminus \|q\|^\sigma_{\mathcal{P},\xi}$;
- $\|Z\|^\sigma_{\mathcal{P},\xi} = \xi(Z)$;
- $\|\phi_1 \wedge \phi_2\|^\sigma_{\mathcal{P},\xi} = \|\phi_1\|^\sigma_{\mathcal{P},\xi} \cap \|\phi_1\|^\sigma_{\mathcal{P},\xi}$;
- $\|\phi_1 \vee \phi_2\|^\sigma_{\mathcal{P},\xi} = \|\phi_1\|^\sigma_{\mathcal{P},\xi} \cup \|\phi_1\|^\sigma_{\mathcal{P},\xi}$;
- $\|\langle A\rangle \mathbf{X}\phi\|^\sigma_{\mathcal{P},\xi} = \{c \in C_\mathcal{P} \mid \exists \upsilon_A : A \to \Theta^\sigma \ s.t.\ \forall \pi \in \mathsf{out}^\sigma(c, \upsilon_A),\ \pi_1 \in \|\phi\|^\sigma_{\mathcal{P},\xi}\}$;
- $\|[A]\mathbf{X}\phi\|^\sigma_{\mathcal{P},\xi} = \{c \in C_\mathcal{P} \mid \forall \upsilon_A : A \to \Theta^\sigma,\ \exists \pi \in \mathsf{out}^\sigma(c, \upsilon_A),\ \pi_1 \in \|\phi\|^\sigma_{\mathcal{P},\xi}\}$;
- $\|\mu Z.\phi\|^\sigma_{\mathcal{P},\xi} = \bigcup\{C \subseteq C_\mathcal{P} \mid \|\phi\|^\sigma_{\mathcal{P},\xi[Z\mapsto C]} \supseteq C\}$;
- $\|\nu Z.\phi\|^\sigma_{\mathcal{P},\xi} = \bigcap\{C \subseteq C_\mathcal{P} \mid \|\phi\|^\sigma_{\mathcal{P},\xi[Z\mapsto C]} \subseteq C\}$;
- $\|\mathbf{K}_i\phi\|^\sigma_{\mathcal{P},\xi} = \{c \in C_\mathcal{P} \mid \forall c' \in C_\mathcal{P}, c \sim_i c' \Longrightarrow c' \in \|\phi\|^\sigma_{\mathcal{P},\xi}\}$;
- $\|\mathbf{E}_A\phi\|^\sigma_{\mathcal{P},\xi} = \{c \in C_\mathcal{P} \mid \forall c' \in C_\mathcal{P}, c \sim^E_A c' \Longrightarrow c' \in \|\phi\|^\sigma_{\mathcal{P},\xi}\}$;
- $\|\mathbf{C}_A\phi\|^\sigma_{\mathcal{P},\xi} = \{c \in C_\mathcal{P} \mid \forall c' \in C_\mathcal{P}, c \sim^C_A c' \Longrightarrow c' \in \|\phi\|^\sigma_{\mathcal{P},\xi}\}$;
- $\|\overline{\mathbf{K}}_i\phi\|^\sigma_{\mathcal{P},\xi} = \{c \in C_\mathcal{P} \mid \exists c' \in C_\mathcal{P}, c \sim_i c' \wedge c' \in \|\phi\|^\sigma_{\mathcal{P},\xi}\}$;
- $\|\overline{\mathbf{E}}_A\phi\|^\sigma_{\mathcal{P},\xi} = \{c \in C_\mathcal{P} \mid \exists c' \in C_\mathcal{P}, c \sim^E_A c' \wedge c' \in \|\phi\|^\sigma_{\mathcal{P},\xi}\}$;
- $\|\overline{\mathbf{C}}_A\phi\|^\sigma_{\mathcal{P},\xi} = \{c \in C_\mathcal{P} \mid \exists c' \in C_\mathcal{P}, c \sim^C_A c' \wedge c' \in \|\phi\|^\sigma_{\mathcal{P},\xi}\}$.

A.2 Proof of Thereom 3

Proof. Given a CEGS $\mathcal{P} = (\mathsf{Ag}, \mathsf{Ac}, P, \Delta, \lambda, \{\sim_i \mid i \in \mathsf{Ag}\})$, we construct a PEGS $\mathcal{P}' = (\mathsf{Ag}, \mathsf{Ac}, P, \Gamma, \Delta', \lambda', \{\sim'_i \mid i \in \mathsf{Ag}\})$, where

- $\Gamma = P \cup \{\perp\}$,
- Δ' is the least function such that for every $\Delta(p, \mathbf{d}) = (p')$, $\Delta'(p, \gamma, \mathbf{d}) = (p', p\gamma)$ for every $\gamma \in \Gamma$,
- $\sim'_i \subseteq C_{\mathcal{P}'} \times C_{\mathcal{P}'}$ such that for every configurations $c = \langle p, p_1 ... p_m\rangle$ and $c' = \langle p', p'_1 ... p'_m\rangle$, $c \sim'_i c'$ iff $p \sim_i p'$ and $p_j \sim_i p'_j$ for all $j \in [m]$.
- λ' is defined as: for every configuration $\langle p, \gamma_1 ... \gamma_m\rangle \in P \times \Gamma^*$, $\lambda'(\langle p, \gamma_1 ... \gamma_m\rangle) = \lambda(p)$.

For every ATL$_{\mathbf{ir}}$/ATL$^*_{\mathbf{ir}}$ ϕ, it is easy to see that $\langle p\rangle \models_{\mathbf{iR}} \phi$ in the CEGS $\mathcal{P}$ iff $\langle p, \perp\rangle \models_{\mathbf{ir}} \phi$ in the PEGS $\mathcal{P}'$. ◻

A.3 Proof of Proposition 3

Proof. i. For every $i \in A$, since $\upsilon_A(i)$ respects to $\approx_i$, i.e., for all $(p, \gamma\omega)$ and $(p', \gamma'\omega')$ with $(p, \gamma) \approx_i (p', \gamma')$, $\upsilon_A(i)(p, \gamma\omega) = \upsilon_A(i)(p', \gamma'\omega')$, we can construct a unique function $\upsilon'_i : P \times \Gamma \to \mathsf{Ac}$ such that $\upsilon'_i(p, \gamma) = \upsilon_A(i)(p, \gamma\omega)$ for some $\omega \in \Gamma^*$. It immediately follows that $\mathsf{out}^{\mathbf{ir}}(c, \upsilon_A) = \mathsf{out}(c, \bigcup_{i \in A} \upsilon'_i)$.

ii. Suppose for every $i \in A$, we have a function $\upsilon'_i : P \times \Gamma \to \mathsf{Ac}$ such that $\upsilon'_i(p, \gamma) = \upsilon'_i(p', \gamma')$ for all (p, γ) and (p', γ') with $(p, \gamma) \approx_i (p', \gamma')$, let $\upsilon_i : C_{\mathcal{P}} \to \mathsf{Ac}$ be the **ir**-strategy such that for every $(p, \gamma) \in P \times \Gamma$, $\omega \in \Gamma^*$, $\upsilon_i(p, \gamma\omega) = \upsilon'_i(p, \gamma)$. It is easy to see that $\mathsf{out}^{\mathbf{ir}}(c, \upsilon_A) = \mathsf{out}(c, \bigcup_{i \in A} \upsilon'_i)$ and for all $(p, \gamma\omega)$ and $(p', \gamma'\omega')$ with $(p, \gamma) \approx_i (p', \gamma')$, $\upsilon_i(p, \gamma\omega) = \upsilon_i(p', \gamma'\omega')$. □

A.4 Proof of Lemma 1

Proof. ($\Longrightarrow$:) Let π be a path which starts from c and π is compatible with respect to υ_A, i.e., $\pi \in \mathsf{out}^{\mathbf{ir}}(c, \upsilon_A)$. We shall show that $\pi \in \prod_{\mathcal{P}_{\upsilon_A}}(c)$. For this purpose, it is sufficient to show that, for all $k \geq 0$, π_{k+1} is a successor of π_k in the PDS $\mathcal{P}_{\upsilon_A}$.

Since π is *compatible* with respect to υ_A, for all $k \geq 0$, there exists some $\mathbf{d}_k \in \mathcal{D}$ such that $\pi_k \overset{\mathbf{d}_k}{\Longrightarrow}_{\mathcal{P}} \pi_{k+1}$ and $\mathbf{d}_k(i) = \upsilon_A(i)(\pi_{\leq k})$ for all $i \in A$. Let $\pi_k = \langle p_k, \omega_k \rangle$ for all $k \geq 0$.

For any $k \geq 0$, there exist $\gamma_k \in \Gamma$, $u_k, u_{k+1} \in \Gamma^*$ such that

- $\omega_k = \gamma_k u_k$,
- $\omega_{k+1} = u_{k+1} u_k$,
- $\langle p_k, \gamma_k \rangle \overset{\mathbf{d}_k}{\hookrightarrow}_{\mathcal{P}} \langle p_{k+1}, u_{k+1} \rangle$.

Since $\mathbf{d}_k(i) = \upsilon_A(i)(\pi_{\leq k})$ for all $i \in A$, according to the construction of $\mathcal{P}_{\upsilon_A}$, we have that $((p_k, \gamma_k), (p_{k+1}, u_{k+1})) \in \Delta'$. Therefore, π_{k+1} is a successor of π_k in the PDS $\mathcal{P}_{\upsilon_A}$.

($\Longleftarrow$:) Let us consider a path $\pi \in \prod_{\mathcal{P}_{\upsilon_A}}(c)$, we shall show that π is compatible with respect to υ_A, i.e., $\pi \in \mathsf{out}^{\mathbf{ir}}(c, \upsilon_A)$. For all $k \geq 0$, π_{k+1} is a successor of π_k in the PDS $\mathcal{P}_{\upsilon_A}$. Let $\pi_k = \langle p_k, \omega_k \rangle$ for all $k \geq 0$.

For any $k \geq 0$, there exist $\gamma_k \in \Gamma$, $u_k, u_{k+1} \in \Gamma^*$ such that

- $\omega_k = \gamma_k u_k$,
- $\omega_{k+1} = u_{k+1} u_k$,
- $((p_k, \gamma_k), (p_{k+1}, u_{k+1})) \in \Delta'$.

According to the construction of $\mathcal{P}_{\upsilon_A}$, there exists a transition $\langle p_k, \gamma_k \rangle \overset{\mathbf{d}_k}{\hookrightarrow}_{\mathcal{P}} \langle p_{k+1}, u_{k+1} \rangle$ in $\mathcal{P}$ such that $\mathbf{d}_k(i) = \upsilon_A(i)(\pi_{\leq k})$ for all $i \in A$. Therefore, π is compatible with respect to υ_A. □

A.5 Proof of Lemma 3

Proof. A configuration $\langle p, [\gamma_k, \boldsymbol{s_k}]...[\gamma_0, \boldsymbol{s_0}] \rangle \in C_{\mathcal{P}'}$ is *consistent* iff for every $i :\ 0 \leq i < k$, $\boldsymbol{\delta}(\boldsymbol{s_i}, \gamma) = \boldsymbol{s_{i+1}}$. Suppose the path of $\mathcal{P}'$ reaches the consistent configuration

$\langle p, [\gamma_k, \boldsymbol{s_k}]...[\gamma_0, \boldsymbol{s_0}]\rangle \in C_{\mathcal{P}'}$, the path of $\mathcal{P}$ is at the configuration $\langle p, \gamma_k...\gamma_0\rangle \in C_{\mathcal{P}}$ and for every $i \in [n]$, the DFA $\mathcal{A}_i$ is at the state $s_{i,k}$ after reading the word $\gamma_0...\gamma_{k-1}$. If there is a transition rule $\langle p, \gamma_k\rangle \overset{\mathbf{d}}{\hookrightarrow}_{\mathcal{P}} \langle p', \epsilon\rangle$, then $\mathcal{P}$ moves from the configuration $\langle p, \gamma_k...\gamma_0\rangle$ to the configuration $\langle p', \gamma_{k-1}...\gamma_0\rangle$ if the agents cooperatively made the decision $\mathbf{d}$. Meanwhile, the DFA $\mathcal{A}_i$ should go to the state $s_{i,k-1}$ after reading the word $\gamma_0...\gamma_{k-2}$, as $\mathcal{A}_i$ is deterministic. These are mimicked by the path of $\mathcal{P}'$ which moves from the configuration $\langle p, [\gamma_k, \mathbf{s}_k]...[\gamma_0, \mathbf{s}_0]\rangle$ to the configuration $\langle p', [\gamma_{k-1}, \boldsymbol{s_{k-1}}]...[\gamma_0, \boldsymbol{s_0}]\rangle$ (c.f. Item 1).

Analogously, if there is a transition rule $\langle p, \gamma_k\rangle \overset{\mathbf{d}}{\hookrightarrow}_{\mathcal{P}} \langle p', \gamma'_t...\gamma'_k\rangle$ for some $t \geq k$, then $\mathcal{P}$ moves from the configuration $\langle p, \gamma_k...\gamma_0\rangle$ to the configuration $\langle p', \gamma'_t...\gamma'_k\gamma_{k-1}...\gamma_0\rangle$ if the agents cooperatively made the decision $\mathbf{d}$. Suppose $\boldsymbol{s_j} \xrightarrow{\gamma'_j} \boldsymbol{s_{j+1}}$ for every $j : \; k \leq j \leq t-1$. To encode the computation information of DFAs, we add the transition rule $\langle p, [\gamma_k, \boldsymbol{s_k}]\rangle \overset{\mathbf{d}}{\hookrightarrow}_{\mathcal{P}'} \langle p', [\gamma'_t, \boldsymbol{s_t}]...[\gamma'_k, \boldsymbol{s_k}]\rangle$ which allows the path of $\mathcal{P}'$ moves from the configuration $\langle p, [\gamma_k, \boldsymbol{s_k}]...[\gamma_0, \boldsymbol{s_0}]\rangle$ to the configuration $\langle p, [\gamma'_t, \boldsymbol{s_t}]...[\gamma'_k, \boldsymbol{s_k}][\gamma_{k-1}, \boldsymbol{s_{k-1}}]...[\gamma_0, \boldsymbol{s_0}]\rangle$.

Therefore, for every configurations $\langle p, \gamma_k...\gamma_0\rangle$, $\langle p', \gamma'_{k'}...\gamma'_0\rangle \in C_{\mathcal{P}}$ and $i \in \mathsf{Ag}$, $\langle p, \gamma_k...\gamma_0\rangle \sim_i \langle p', \gamma'_{k'}...\gamma'_0\rangle$ iff there are two consistent configurations $\langle p, [\gamma_k, \boldsymbol{s_k}]...[\gamma_0, \boldsymbol{s_0}]\rangle$, $\langle p', [\gamma'_{k'}, \boldsymbol{s'_{k'}}]...[\gamma'_0, \boldsymbol{s'_0}]\rangle \in C_{\mathcal{P}'}$ such that $\langle p, [\gamma_k, \boldsymbol{s_k}]...[\gamma_0, \boldsymbol{s_0}]\rangle \sim'_i \langle p', [\gamma'_{k'}, \boldsymbol{s'_{k'}}]...[\gamma'_0, \boldsymbol{s'_0}]\rangle$. Since the automata $\mathcal{A}'_i s$ are deterministic, all the reachable configurations starting from consistent configurations are consistent configurations. The result immediately follows. □

A.6 Proof of Lemma 4

Proof. Consider a PEGS $\mathcal{P} = (\mathsf{Ag}, \mathsf{Ac}, P, \Gamma, \Delta, \lambda, \{\sim_i \mid i \in \mathsf{Ag}\})$ and a closed AEMC$_\mathbf{i}$ formula ϕ, we shall show how to construct an AMA $\mathcal{A}_\phi$ such that $||\phi||^{\mathbf{i}}_{\mathcal{P}} = \mathcal{L}(\mathcal{A}_\phi)$. The AMA $\mathcal{A}_\phi$ is defined based on the structure of ϕ. The main framework follows Procedure 1 of [36].

We only need to give procedures for subformulae of the form $\langle A\rangle\mathbf{X}\phi'$, $[A]\mathbf{X}\phi'$, $\mathbf{K}_i\phi'$, $\overline{\mathbf{K}}_i\phi'$, $\mathbf{E}_A\phi'$, $\mathbf{C}_A\phi'$, $\overline{\mathbf{E}}_A\phi'$ or $\overline{\mathbf{C}}_A\phi'$. Suppose the size-preserving EARs $\sim_i$ for $i \in \mathsf{Ag}$ are specified by equivalence relations $\simeq_i \subseteq (P \times P) \cup (\Gamma \times \Gamma)$.

- For ψ of the form $\langle A\rangle\mathbf{X}\phi'$ (resp. $[A]\mathbf{X}\phi'$), we use the approach of [13]. Let $succ_f(p, \gamma)$ denote the set of tuples $\{\langle p', \omega\rangle \mid \langle p, \gamma\rangle \overset{\mathbf{d}}{\hookrightarrow}_{\mathcal{P}} \langle p', \omega\rangle \in \Delta$ and $\forall a \in Domain(f) : \mathbf{d}(a) = f(a)\}$, and $succ_f(p, \gamma\omega')$ denote the set of configurations $\{\langle p', \omega\omega'\rangle \mid \langle p', \omega\rangle \in succ_f(p, \gamma)\}$ for every $\omega' \in \Gamma^*$ which is the set of all the possible successors of $\langle p, \gamma\omega'\rangle$ on the actions $f(a)$ for $a \in Domain(f)$ (agents $\mathsf{Ag} \setminus A$ can make any action).
  Suppose the AMA $\mathcal{A}_{\phi'} = (S_1, \Gamma, \delta_1, I_1, S^1_f)$ has been constructed using the Dispatch of [36]. We construct the AMA $\mathcal{A}_\psi = (S_1 \cup I, \Gamma, \delta_1 \cup \delta', I, S^1_f)$ where $I = \{[p, \psi] \mid p \in P\}$, and

$$\delta'([p, \psi], \gamma) = \bigvee_{f:A\to Ac} \bigwedge_{\langle p', \omega\rangle \in succ_f(p,\gamma)} \bigwedge_{s \in Q_{p'\omega}} s$$

(resp.

$$\delta'([p,\psi],\gamma) = \bigwedge_{f:A\to Ac} \bigvee_{\langle p',\omega\rangle \in succ_f(p,\gamma)} \bigwedge_{s\in Q_{p'\omega}} s\ )$$

where $p' \xrightarrow{\omega}_{\delta_1} Q_{p'\omega}$.
The correctness follows from the fact that at a configuration $\langle p,\omega\rangle$, each agent in A only has to select one action and it can forget this selection and select a new action at next time. Therefore, the selection can be represented by function $f : A \to Ac$.

- For ψ of the form $\mathbf{K}_i\phi'$, suppose the AMA $\mathcal{A}_{\phi'} = (S', \Gamma, \delta', I', S'_f)$ has been constructed using Dispatch of [36]. We construct $\mathcal{A}_\psi = (S', \Gamma, \delta, I, S'_f)$ where
  - $I = \{p \in P \mid \exists p' \in I'.\ p \simeq_i p'\}$.
  - For each $(p,\gamma) \in P \times \Gamma$, let $[p]_{\simeq_i}$ (resp. $[\gamma]_{\simeq_i}$) be the equivalence of p (resp. γ) under $\simeq_i$, and $\overline{S'_{p,\gamma}} := \{S'_{p,\gamma} \mid (p,\gamma,S'_{p,\gamma}) \in \delta'\}$. Then $(p,\gamma,S) \in \delta$ if (1) for all $p' \in [p]_{\simeq_i}$ and $\gamma' \in [\gamma]_{\simeq_i}$, $\overline{S'_{p',\gamma'}} \neq \emptyset$; and (2) $S = \bigcup_{p'\in[p]_{\simeq_i},\gamma'\in[\gamma]_{\simeq_i}} S''_{p',\gamma'}$, where $S''_{p',\gamma'} \in \overline{S'_{p',\gamma'}}$.
  - For every $(s,\gamma,S) \in \delta'$ such that $s \in S' \setminus P$, let $(s,\gamma',S) \in \delta$ for every $\gamma' \in \Gamma$ with $\gamma' \simeq_i \gamma$.
- For ψ of the form $\overline{\mathbf{K}}_i\phi'$, suppose the AMA $\mathcal{A}_{\phi'} = (S', \Gamma, \delta', I', S'_f)$ has been constructed using Dispatch of [36]. We construct $\mathcal{A}_\psi = (S', \Gamma, \delta, I, S'_f)$ where
  - $I = \{p \in P \mid \exists p' \in I'.\ p \sim_i p'\}$.
  - For each $(p,\gamma) \in P\times\Gamma$, if there is $(p',\gamma',S'_1) \in \delta'$ such that $p \simeq_i p'$ and $\gamma \simeq_i \gamma'$, let $(p,\gamma,S'_1) \in \delta$.
  - For every $(s,\gamma,S) \in \delta'$ such that $s \in S' \setminus P$, let $(s,\gamma',S) \in \delta$ for every $\gamma' \in \Gamma$ with $\gamma' \simeq_i \gamma$.
- For ψ of the form $\mathbf{E}_A\phi'$, $\mathbf{C}_A\phi'$, $\overline{\mathbf{E}}_A\phi'$ or $\overline{\mathbf{C}}_A\phi'$, $\mathcal{A}_\psi$ can be constructed in a very similar way, in which the relation $\simeq_i$ is replaced by the relation $\bigcup_{i\in A} \simeq_i$ (resp. the transitive closure of $\bigcup_{i\in A} \simeq_i$). The correctness immediately follows from the above constructions.

Note that in [36], an integer that broadly corresponds to the fixed point depth of ψ in ϕ was added into $[p,\psi]$. For the sake of simplification, we omitted this integer in the above construction.

□