ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230439

Epistemic JAADL: A Modal Logic for Joint Abilities with Imperfect Information

Zhaoshuai Liu[a], Aiting Liang[a] and Yongmei Liu[a;*]

[a]Dept. of Computer Science, Sun Yat-sen University, Guangzhou 510006, China
ORCiD ID: Zhaoshuai Liu https://orcid.org/0000-0002-8508-5482,
Aiting Liang https://orcid.org/0000-0002-1523-2627, Yongmei Liu https://orcid.org/0000-0003-2039-7626

Abstract. Coordination and joint ability are important problems in representation and reasoning about multi-agent systems. Ghaderi *et al.* presented a formalization of joint ability of coalitions in the expressive first-order language of the situation calculus. Essentially, a coalition has joint ability to achieve a goal if after iterated elimination of dominated strategies, any remaining joint strategy achieves the goal. Based on their work, Liu *et al.* proposed JAADL, a modal logic for joint abilities under strategy commitments. In this paper, we propose EJAADL, an epistemic extension of JAADL, for imperfect information games where agents may have incomplete knowledge or even false beliefs about the world. Like Ghaderi *et al.*'s work, elimination of dominated strategies is now based on beliefs about the world, rather than facts about the world as in JAADL. Strategies are required to be uniform, *i.e.*, they select the same action in all accessible histories. We illustrate EJAADL with examples, analyze its properties, and show that model checking memoryless EJAADL is in EXPTIME. Moreover, we consider the fragment of EJAADL without the iterated elimination operator, and show that model-checking the memoryless version of this fragment can be done in PSPACE.

1 Introduction

Representation and reasoning about strategic abilities has been an active research area in AI and multi-agent systems. The foundational work is Alternating-time Temporal Logic ATL/ATL* [1] where formula $\langle\langle A\rangle\rangle\phi$ expresses that coalition A has a group strategy to ensure temporal goal ϕ holds no matter what the other agents do. However, strategies are treated implicitly in ATL. Treating strategies as explicit first-order objects, Mogavero *et al.* proposed Strategy Logic SL [19], a very expressive logic for strategic reasoning that strictly contains ATL* and can express many game-theoretic notions such as existence of Nash equilibria. However, in reality, players often have imperfect information about the game states. To deal with imperfect information games, Alternating-time Temporal Epistemic Logic (ATEL) and its variations have been proposed [13, 14, 15, 9, 5]. Recently, SL has been extended to take into account of imperfect information [8, 2, 4, 6] and allow for epistemic reasoning under the uninformed or informed semantics [18, 3], depending on whether agents know other agents' strategies. EGDL [16] also concerns representing and reasoning about imperfect information games: it has the standard epistemic operators, but without the until or coalition operators, and provides an imperfect recall semantics, thus reducing the model checking complexity. In these logics, imperfect information is captured by equipping the models with accessibility relations which are usually equivalence relations. Strategies are then required to be uniform, *i.e.*, they select the same action in all indistinguishable histories.

Coordination and joint ability are important problems in representation and reasoning about multi-agent systems. "A team of agents is jointly able to achieve a goal if despite any incomplete knowledge or even false beliefs that they may have about the world or each other, they still know enough to be able to get to a goal state, should they choose to do so" [11]. Many strategic logics ignore the coordination problem: a coalition may have many group strategies to ensure a goal, yet a player may not know others' choices, hence the coalition may end up with a group strategy which may not ensure the goal.

Nonetheless, there have been works on developing logical theories of coordination and joint abilities. Ghaderi *et al.* studied the coordination problem for imperfect information games where agents may have incomplete knowledge or even false beliefs about the world, and presented a formalization of joint ability of coalitions [11] based on the idea of iterated elimination of dominated strategies [20]. Essentially, a coalition has joint ability to achieve a goal if after iterated elimination of dominated strategies, any remaining joint strategy achieves the goal. However, their work uses the very expressive situation calculus [22], making it difficult to explore computational properties of the logic. Based on coalition logic [21], Hawke proposed a logic of joint ability in two-player tacit games with a joint ability modality $((A))\varphi$ [12]: two players have joint ability to achieve a goal if after elimination of punishment strategies, *i.e.*, those strategies that fail to achieve the goal no matter what other agents do, any remaining joint strategy achieves the goal. So they only eliminate punishment strategies, hence their concept of joint ability is much weaker than that of [11]. Recently, based on Ghaderi *et al.*'s idea, Liu *et al.* proposed JAADL [17], a modal logic for joint abilities under strategy commitments, which extends ATL*. Firstly, they introduce an operator $(A)^{\infty}_{\psi}\varphi$, meaning φ holds after iterated elimination of dominated strategies w.r.t. group A and goal ψ, with which they can represent joint abilities of coalitions. Secondly, their logic is based on linear dynamic logic LDL [23], so that they can use regular expressions to represent commitments to structured strategies. However, unlike Ghaderi *et al.*'s work, JAADL is for perfect information games.

In this paper, we propose EJAADL, an epistemic extension of JAADL, for imperfect information games where agents may have incomplete knowledge or even false beliefs about the world. To model

* Corresponding Author. Email: ymliu@mail.sysu.edu.cn

beliefs, we equip the models with accessibility relations that are serial, Euclidean, and transitive. Strategies are then required to be uniform in the sense that they select the same action in all accessible histories. Like Ghaderi *et al.*'s work, elimination of dominated strategies (EDS) is now based on beliefs about the world, rather than facts about the world as in JAADL. In particular, an agent eliminates a strategy if she *believes* there is a better strategy; an agent i believes among her strategies, σ' is better than σ if for each group strategy σ_{-i} that i *believes* the other agents keep, σ works with σ_{-i} to achieve the goal implies σ' does too. Note that the strategies agent i keeps might be different from those agent j believes agent i keeps. We illustrate the syntax and semantics of EJAADL with examples from the literature. Then we analyze properties of EJAADL. We show that two sufficient conditions and one necessary condition for joint abilities in JAADL can be modified to hold in EJAADL. Finally, we show that model checking memoryless EJAADL can be solved in EXPTIME. Moreover, we consider the fragment of EJAADL without the iterated elimination operator, and show that model-checking the memoryless version of this fragment can be done in PSPACE.

2 Preliminaries

In this section, we introduce JAADL. We begin with the concepts of concurrent game structures and strategies.

Let AP be a finite non-empty set of atoms, AC a finite non-empty set of actions, and let $AG = \{1, \dots, n\}$ be a finite non-empty set of agents. We use $\emptyset$ to denote the empty set.

Definition 1 (Concurrent Game Structures) A concurrent game structure (CGS) is a tuple $\mathcal{G} = \langle W, L, P, \tau, w^0 \rangle$, where

- W is a finite non-empty set of states; $w^0 \in W$ is a designated initial state; L is a labeling function mapping each state to a subset of AP; τ is a transition function mapping a state w and a decision at w to a new state;
- for each agent i, P_i is a possible action function mapping each state to a subset of AC; a decision at state w is a function mapping each agent i to an action from $P_i(w)$; we use $D(w)$ to denote the set of decisions at w;

Example 1 (Robots and Carriage from [7]) As shown in Figure 1 (we ignore the dotted lines for now), each robot can either push or wait. Moreover, they both use the same force when pushing. Thus, if the robots push simultaneously or wait simultaneously, the carriage does not move. When only one of the robots is pushing, the carriage moves accordingly. We formalize the example as a CGS $\mathcal{G}$:

- $AG = \{1, 2\}$, $AP = \{pos_0, pos_1, pos_2\}$, $AC = \{push, wait\}$, $W = \{q_0, q_1, q_2\}$, $w^0 = q_0$;
- $L(q_0) = \{pos_0\}$, $L(q_1) = \{pos_1\}$, $L(q_2) = \{pos_2\}$;
- $P_i(w) = AC$ for each $i \in AG$ and each $w \in W$;
- τ is described directly in Figure 1.

We now define tracks and paths. Tracks (resp., path) are finite (resp., infinite) state-decision sequences, they are used to define strategies (resp., interpret path formulas).

Definition 2 A track h in a CGS $\mathcal{G}$ is a finite state-decision sequence $w_0 d_0 w_1 d_1 \dots w_k$ s.t. for all i ($0 \le i < k$), $d_i \in D(w_i)$, and $w_{i+1} = \tau(w_i, d_i)$. We use $last(h)$ to denote w_k.

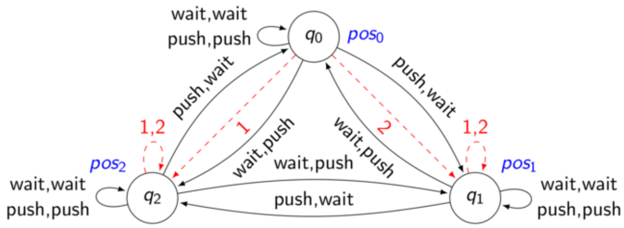

Figure 1. Robots and Carriage

Definition 3 A path λ in a CGS $\mathcal{G}$ is an infinite state-decision sequence $w_0 d_0 w_1 d_1 \dots$ s.t. for all $i \ge 0$, $d_i \in D(w_i)$, and $w_{i+1} = \tau(w_i, d_i)$.

Definition 4 A strategy for agent i starting from state w is a function mapping each track h beginning from w to an action from $P_i(last(h))$. We let $Str_i(w)$ denote the set of all strategies for agent i starting from w.

To handle elimination of strategies, Liu *et al.* introduce the concept of strategy spaces. A strategy space specifies the set of possible strategies for each agent [17].

Definition 5 A strategy space s starting from state w is a function mapping each agent i to a subset of $Str_i(w)$. The full strategy space $\text{fs}(w)$ starting from state w maps each agent i to $Str_i(w)$.

For a group A of agents, s_A means the restriction of s to group A.

Definition 6 A memoryless strategy for agent i is a function mapping each state w to an action from $P_i(w)$. The full memoryless strategy space, denoted by *fms*, maps each agent i to the set of all memoryless strategies for i.

We use σ to range over strategies. A group strategy of $A \subseteq AG$ is a mapping from A to strategies. We use σ_A to range over group strategies of A, σ_i to range over strategies for agent i. We use $-A$ to denote $AG - A$. For $i \in AG$, we use $-i$ to denote $AG - \{i\}$. A joint strategy is a group strategy of AG. We use σ_{all} to range over joint strategies. Thus (σ_A, σ_{-A}) stands for the joint strategies composed of σ_A and σ_{-A}.

Definition 7 A state w and a joint strategy σ_{all} determine a unique path $w_0 d_0 w_1 d_1 w_2 d_2 \dots$ as follows: $w_0 = w$, and for each $j \ge 0$, d_j is the decision associated to the track $w_0 \dots w_j$, *i.e.*, for each agent i, $d_j(i) = \sigma_i(w_0 \dots w_j)$, and $w_{j+1} = \tau(w_j, d_j)$. We use $out(w, \sigma_{all})$ to denote this path.

We begin with the syntax of JAADL. We use φ to denote state formulas, ψ path formulas, ϕ propositional formulas, and ρ path expressions, which are regular expressions over propositional formulas and tests of path formulas. Other than atomic propositions from AP, there are atomic propositions of the form a_i where $a \in$ AC and $i \in AG$, meaning agent i does action a. We use $\top$ to denote $true$.

Definition 8 JAADL formulas are built as follows:

$$\varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle\!\langle A \rangle\!\rangle \psi \mid (A)_\psi \varphi \mid (A)^\infty_\psi \varphi$$

$$\psi ::= \varphi \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \langle \rho \rangle \psi$$

$$\rho ::= \phi \mid \psi? \mid \rho_1 + \rho_2 \mid \rho_1;\rho_2 \mid \rho^*$$
$$\phi ::= p \mid a_i \mid \neg\phi \mid \phi_1 \wedge \phi_2$$

where $p \in AP$, and $A \subseteq AG$.

Intuitively, $\langle\rho\rangle\psi$ means that from the current state in the path there exists an execution satisfying the path expression ρ such that its last state satisfies ψ. We use $[\rho]\psi$ as abbreviation for $\neg\langle\rho\rangle\neg\psi$.

Intuitively, $\langle\langle A\rangle\rangle\psi$ means group A has a strategy to achieve ψ. We usually write $\langle\langle i_1,\ldots,i_k\rangle\rangle$ instead of $\langle\langle\{i_1,\ldots,i_k\}\rangle\rangle$ where $i_1,\ldots,i_k \in AG$. For a special case, $\langle\langle\varnothing\rangle\rangle\psi$ means ψ holds no matter how the agents play. $(A)_\psi\varphi$ (resp., $(A)^\infty_\psi\varphi$) means φ holds after one step (resp., iterated) elimination of dominated strategies w.r.t. group A and the goal ψ. We use $(A)^2_\psi\varphi$ to denote $(A)_\psi(A)_\psi\varphi$, and similarly for $(A)^k_\psi\varphi$, where $k \in \mathbb{N}$.

We use $((A))^k\psi$ to abbreviate for $(A)^k_\psi\langle\langle\varnothing\rangle\rangle\psi$. When $k = 1$, we simply write $((A))\psi$. Intuitively, $((A))^k\psi$ means after k-round elimination of dominated strategies, ψ holds no matter how the agents play, and we say group A has stage k joint ability to achieve ψ. We use $((A))^\infty\psi$ to abbreviate for $(A)^\infty_\psi(\langle\langle A\rangle\rangle\psi \wedge \langle\langle\varnothing\rangle\rangle\psi)$. Here we need $\langle\langle A\rangle\rangle\psi$ because when the strategy space is infinite, it might become empty after iterative EDS, as illustrated by Example 3 from [17]. Intuitively, $((A))^\infty\psi$ means group A has joint ability to ensure ψ, *i.e.*, after iterative EDS, ψ holds no matter how the agents play.

We now provide the semantics of JAADL. We begin with the semantics of propositional formulas, which are interpreted over state-decision pairs.

Definition 9 Given a CGS $\mathcal{G}$, a state w, and a decision d at w, we interpret propositional formulas (we omit the cases of $\neg$ and $\wedge$) inductively:

- $w, d \vDash p$ if $p \in L(w)$;
- $w, d \vDash a_i$ if $d(i) = a$;

When interpreting state formulas w.r.t. a strategy space, we make use of two operators on strategy spaces: $R_{A,\psi,w}(s)$ (resp., $R^\infty_{A,\psi,w}(s)$) means the reduction of s via one step (resp., iterated) elimination of dominated strategies.

Definition 10 (JAADL Semantics) Given a CGS $\mathcal{G}$, a state w, a strategy space s, and a path λ, we interpret state formulas and path formulas (we omit the cases of $\neg$ and $\wedge$) and define the operators $R_{A,\psi,w}(s)$ and $R^\infty_{A,\psi,w}(s)$ inductively:

- $w, s \vDash p$ if $p \in L(w)$.
- $w, s \vDash \langle\langle A\rangle\rangle\psi$ if there exists a group strategy $\sigma_A \in s_A$ such that for all strategies $\sigma_{-A} \in s_{-A}$, we have $out(w, (\sigma_A, \sigma_{-A})), s \vDash \psi$.
- $w, s \vDash (A)_\psi\varphi$ if $w, R_{A,\psi,w}(s) \vDash \varphi$.
- $w, s \vDash (A)^\infty_\psi\varphi$ if $w, R^\infty_{A,\psi,w}(s) \vDash \varphi$.
- $\lambda, s \vDash \varphi$ if $w_0, s \vDash \varphi$, where $\lambda = w_0 d_0 w_1 \ldots$.
- $\lambda, s \vDash \langle\phi\rangle\psi$ if $w_0, d_0 \vDash \phi$ and $\lambda', s \vDash \psi$, where $\lambda = w_0 d_0 w_1 \ldots$ and $\lambda' = w_1 d_1 \ldots$.
- $\lambda, s \vDash \langle\psi_1?\rangle\psi_2$ if $\lambda, s \vDash \psi_1$ and $\lambda, s \vDash \psi_2$.
- $\lambda, s \vDash \langle\rho_1 + \rho_2\rangle\psi$ if $\lambda, s \vDash \langle\rho_1\rangle\psi$ or $\lambda, s \vDash \langle\rho_2\rangle\psi$.
- $\lambda, s \vDash \langle\rho_1;\rho_2\rangle\psi$ if $\lambda, s \vDash \langle\rho_1\rangle\langle\rho_2\rangle\psi$.
- $\lambda, s \vDash \langle\rho^0\rangle\psi$ if $\lambda, s \vDash \psi$.
- $\lambda, s \vDash \langle\rho^{k+1}\rangle\psi$ if $\lambda, s \vDash \langle\rho^k;\rho\rangle\psi$ for $k \in \mathbb{N}$.
- $\lambda, s \vDash \langle\rho^*\rangle\psi$ if there exists $k \in \mathbb{N}$ such that $\lambda, s \vDash \langle\rho^k\rangle\psi$.

For $\sigma_i \in s_i$, we define the set of strategies of $-i$ that work with σ_i to ensure ψ w.r.t. state w and strategy space s as follows: $M_{\psi,w,s}(\sigma_i) = \{\sigma_{-i} \in s_{-i} \mid out(w, (\sigma_i, \sigma_{-i})), s \vDash \psi\}$. For $\sigma_i, \sigma'_i \in s_i$, we write $\sigma_i \geq_{\psi,w,s} \sigma'_i$ if $M_{\psi,w,s}(\sigma_i) \supseteq M_{\psi,w,s}(\sigma'_i)$, and we say σ_i weakly dominates σ'_i; we write $\sigma_i >_{\psi,w,s} \sigma'_i$ if $M_{\psi,w,s}(\sigma_i) \supset M_{\psi,w,s}(\sigma'_i)$, and we say σ_i dominates σ'_i.

For a strategy space s, we define the reduction of s w.r.t. group A, goal ψ and state w: $R_{A,\psi,w}(s) = s'$ s.t. if $i \notin A$, $s'_i = s_i$; otherwise, $s'_i = \{\sigma_i \in s_i \mid \neg\exists\sigma'_i \in s_i.\sigma'_i >_{\psi,w,s} \sigma_i\}$. For $k \geq 2$, we define $R^k_{A,\psi,w}(s) = R_{A,\psi,w}(R^{k-1}_{A,\psi,w}(s))$. Finally, we define the iterative reduction of s: $R^\infty_{A,\psi,w}(s) = s'$ s.t. for $i \in AG$, $s'_i = \bigcap^\infty_{k=0} R^k_{A,\psi,w}(s)_i$.

Note that the comparison of σ_i and σ'_i at w is based on facts about w. Since we consider perfect information games, each agent in A is able to compute the k round elimination of dominated strategies for herself and each other agent in A.

Definition 11 A state formula φ is valid if for all CGS $\mathcal{G}$, we have $\mathcal{G} \vDash \varphi$, meaning $w^0, \mathrm{fs}(w^0) \vDash \varphi$, where w^0 is the initial state of $\mathcal{G}$.

Recall $\mathrm{fs}(w)$ is the full strategy space starting from state w.

Example 1 cont'd. We have $\mathcal{G} \vDash ((1,2))\langle\top\rangle pos_1$, meaning that agents 1 and 2 have stage-1 joint ability to bring the carriage to pos_1 in the state. Since the goal $\langle\top\rangle pos_1$ only concerns the next state, at q_0, each agent has two strategies: $push$ and $wait$. The first table of Figure 3 shows whether each joint strategy can achieve the goal. Thus for agent 1, $push$ dominates $wait$, which is eliminated; for agent 2, $wait$ dominates $push$, which is eliminated. The only remaining joint strategy achieves the goal. Hence there is stage-1 joint ability.

We now introduce some terminology about strategies, which are used in analyzing properties of JAADL.

Definition 12 We say σ is a winning strategy for i w.r.t. ψ, w, s if $M_{\psi,w,s}(\sigma)$ is s_{-i}. We say σ is a punishment strategy for i w.r.t. ψ, w, s if $M_{\psi,w,s}(\sigma) = \varnothing$. We say σ is an optimal strategy for i w.r.t. ψ, w, s, if for any $\sigma' \in s_i$, $\sigma \geq_{\psi,w,s} \sigma'$.

Thus σ is a winning strategy for i if σ works with any strategy of $-i$. The formula $\langle\langle i\rangle\rangle\psi$ represents that i has a winning strategy w.r.t. goal ψ. Similarly, σ is a punishment strategy for i if σ works with no strategy of $-i$. The formula $\langle\langle i\rangle\rangle\neg\psi$ expresses that i has a punishment strategy w.r.t. goal ψ.

Definition 13 We say that two strategies σ and σ' are equivalent w.r.t. ψ, w, s if $M_{\psi,w,s}(\sigma) = M_{\psi,w,s}(\sigma')$. We say that two strategies σ and σ' are incomparable w.r.t. ψ, w, s if $M_{\psi,w,s}(\sigma) \nsubseteq M_{\psi,w,s}(\sigma')$ and $M_{\psi,w,s}(\sigma') \nsubseteq M_{\psi,w,s}(\sigma)$.

Definition 14 Given a CGS $\mathcal{G}$, a state w, a strategy space s, and a goal ψ, the payoff matrix for ψ, denoted C_ψ is a 0-1 matrix defined as follows: for each $\sigma_{all} \in s$, $C_\psi(\sigma_{all}) = 1$ iff $out(w, \sigma_{all}), s \vDash \psi$.

Intuitively, in the 0-1 matrix, the payoff for a joint strategy is 1 if it achieves the goal, and 0 otherwise.

3 Syntax and Semantics of EJAADL

In this section, we propose EJAADL, an epistemic extension of JAADL, for imperfect information games where agents may have incomplete knowledge or even false beliefs about the world. Like in Ghaderi *et al.*'s work, elimination of dominated strategies is now based on beliefs about the world, rather than facts about the world as in JAADL.

First, we introduce an epistemic extension of concurrent game structures to model imperfect information games where agents may

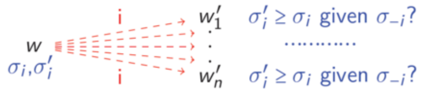

Figure 2. Strategy comparison according to beliefs

have false beliefs. Note that unlike in usual epistemic extensions, we require the accessibility relations to be KD45 relations rather than equivalence relations.

Definition 15 An epistemic concurrent game structure (ECGS) is a CGS $\mathcal{G} = \langle W, L, P, \tau, w^0 \rangle$ extended with $\to$, where for each agent i, $\to_i$ is a binary relation on W that is serial, Euclidean, and transitive.

Example 2 (Robots and Carriage with imperfect information) Figure 1 shows an ECGS where the dotted lines represent the accessibility relations. Intuitively, agent 1 mistakes q_0 as q_2, and agent 2 mistakes q_0 as q_1; but both agents have complete information about q_1 and q_2. It is easy to check that the accessibility relations are serial, Euclidean, and transitive.

The $\to_i$ relation on states can be extended to histories:

Definition 16 Let $h = w_0 d_0 w_1 d_1 \ldots w_k$ and $h' = w'_0 d'_0 w'_1 d'_1 \ldots w'_k$ be tracks of the same length. We write $h \to_i h'$, if for each $j \leq k$, we have $w_j \to_i w'_j$ and $d_j = d'_j$.

It is easy to show that the $\to_i$ relation on histories is Euclidean and transitive, but not serial. For the Robots and Carriage example, we have $q_0(w,w)q_0 \to_1 q_2(w,w)q_2$.

Strategies are now required to be uniform in the sense that they select the same action in all accessible histories.

Definition 17 A strategy σ for agent i is uniform if when $h_1 \to_i h_2$, then $\sigma(h_1) = \sigma(h_2)$. We let $\mathrm{Ustr}_i(w)$ denote the set of all uniform strategies for agent i starting from w. A memoryless strategy σ for agent i is uniform if when $w_1 \to_i w_2$, then $\sigma(w_1) = \sigma(w_2)$.

Since the accessibility relations on states or histories are Euclidean and transitive, the above notions of uniform strategies are well-defined.

Since elimination of dominated strategies is now based on beliefs, we need to extend the concept of strategy space from JAADL. In JAADL, in order to decide if group A has joint ability to achieve ψ at state w, each agent in A first considers all her possible strategies from w, and then in an iterative manner, each agent in A compares the strategies she keeps at w and eliminates those dominated ones given what strategies the other agents keep. Thus in JAADL, for a state w, a strategy space specifies the set of possible strategies for each agent. However, in EJAADL, at a state w, each agent in A compares the strategies she keeps at w and eliminates those she *believes* are dominated given her *beliefs* about what strategies the other agents keep. So as shown in Figure 2, an agent i in A has to compare two strategies σ_i and σ'_i she keeps at w, *i.e.*, consider each state w' accessible from w, check if $\sigma'_i \geq \sigma_i$ at w', *i.e.*, for each group strategy σ_{-i} the other agents keep at w', check if whenever σ_i works with σ_{-i} to achieve the goal at w', so does σ'_i. Thus, in EJAADL, a strategy space specifies the set of possible strategies for each agent and each state.

Definition 18 A uniform strategy space s is a function mapping each agent i and state w to a subset of $\mathrm{Ustr}_i(w)$. The full uniform strategy space fus maps each (i,w) to $\mathrm{Ustr}_i(w)$.

Let s be a uniform strategy space. Let $i \in AG$, $A \subseteq AG$, and w a state. We use $s_{i,w}$ to denote $s(i,w)$. We use $s_{A,w}$ to denote the set of σ_A s.t. for each $i \in A$, $\sigma_i \in s_{i,w}$. We will simply write s_w for $s_{AG,w}$.

We now present the syntax and semantics of EJAADL.

Definition 19 (EJAADL Syntax) EJAADL adds to JAADL two belief constructors $\varphi ::= B_i\varphi$, and $\psi ::= B_i\psi$, where $i \in AG$, φ is a state formula, and ψ is a path formula.

Intuitively, $B_i\varphi$ (resp., $B_i\psi$) means agent i believes φ at a state (resp., path), We let $\hat{B}_i$ stand for the dual operator of B_i, *i.e.*, $\hat{B}_i\phi$ stand for $\neg B_i \neg\phi$, where ϕ is a state or path formula. Intuitively, $\hat{B}_i\phi$ means agent i believes it is possible that ϕ.

Definition 20 (EJAADL Semantics) Given an ECGS $\mathcal{G}$, a state w, a uniform strategy space s, and a joint uniform strategy $\sigma_{all} \in s_w$, we interpret state formulas and path formulas (we omit the cases of $\neg$ and $\wedge$) and define the operators $R_{A,\psi}(s)$ and $R^{\infty}_{A,\psi}(s)$ inductively:

- $w, s \vDash p$ if $p \in L(w)$.
- $w, s \vDash \langle\!\langle A \rangle\!\rangle \psi$ if there exists a uniform group strategy $\sigma_A \in s_{A,w}$ such that for all uniform strategies $\sigma_{-A} \in s_{-A,w}$, we have $w, (\sigma_A, \sigma_{-A}), s \vDash \psi$.
- $w, s \vDash (A)_\psi \varphi$ if $w, R_{A,\psi}(s) \vDash \varphi$.
- $w, s \vDash (A)^{\infty}_{\psi} \varphi$ if $w, R^{\infty}_{A,\psi}(s) \vDash \varphi$.
- $w, s \vDash B_i\varphi$ if for all w' s.t. $w \to_i w'$, $w', s \vDash \varphi$.
- $w, \sigma_{all}, s \vDash B_i\psi$ if for all w' s.t. $w \to_i w'$, for all $\sigma'_{all} \in s_{w'}$ s.t. $\sigma_i = \sigma'_i$, $w', \sigma'_{all}, s \vDash \psi$.
- for the other cases of path formulas, the interpretation is similar as that in JAADL.

For a uniform strategy space s, we define the reduction of s w.r.t. group A and goal ψ as follows: $R_{A,\psi}(s) = s'$ s.t. for any state w, if $i \notin A$, then $s'_{i,w} = s_{i,w}$; otherwise,

$$s'_{i,w} = \{\sigma_i \in s_{i,w} \mid \neg\exists \sigma'_i \in s_{i,w}. B_i(\sigma'_i >_{\psi,w,s} \sigma_i)\},$$

where $B_i(\sigma'_i >_{\psi,w,s} \sigma_i)$ represents

$$[\forall w' \text{ s.t. } w \to_i w'. \sigma'_i \geq_{\psi,w',s} \sigma_i] \wedge$$
$$[\exists w' \text{ s.t. } w \to_i w' \text{ and } \sigma'_i >_{\psi,w',s} \sigma_i],$$

here $\sigma'_i \geq_{\psi,w',s} \sigma_i$ means

$$\forall \sigma_{-i} \in s_{-i,w'}, \text{ if } w', (\sigma_i, \sigma_{-i}), s \vDash \psi, \text{ then } w', (\sigma'_i, \sigma_{-i}), s \vDash \psi.$$

$R^{\infty}_{A,\psi}(s)$ is similarly defined as in JAADL.

Intuitively, $B_i(\sigma'_i >_{\psi,w,s} \sigma_i)$ means agent i believes that σ'_i dominates σ_i at state w, *i.e.*, for all w' accessible from w by i, σ'_i weakly dominates σ_i at w', and there exists a w' accessible from w by i s.t. σ'_i dominates σ_i at w'.

The interpretations of $B_i\varphi$ and $B_i\psi$ deserve some explanations. $B_i\varphi$ (resp., $B_i\psi$) is a state (resp., path) formula and it is evaluated w.r.t. a state (resp., a state and a joint strategy σ_{all}). Here we assume an agent does not know other agents' strategies. So we are adopting the uninformed semantics in a strategic context [18]. We will give an example of $B_i\psi$ formulas in Example 2.

Below is an easy property of $R_{A,\psi}(s)$.

	1\2	p	w
q_0	p	×	✓
	w	×	×

	1\2	p	w
q_1	p	✓	×
	w	×	✓

	1\2	p	w
q_2	p	×	×
	w	✓	×

Figure 3. Payoff matrices for Robots and Carriage

Round 1	
$q_0, q_2, 1$:	w
$q_1, 1$:	p, w
$q_0, q_1, 2$:	p, w
$q_2, 2$:	p

Round 2		
no change		$\Longrightarrow$ no joint ability
$q_0, 1$:	w	
$q_0, 2$:	p, w	

Figure 4. Iterative EDS for Robots and Carriage

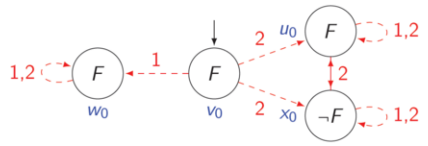

Figure 5. The initial beliefs of agents for Examples 3 and 4

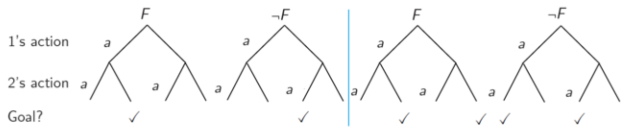

Figure 6. Goals for Examples 3 (left) and 4 (right)

Proposition 1 *Let $i \in AG$, $w \to_i w'$, and $s_{i,w} = s_{i,w'}$. Let $s' = R_{A,\psi}(s)$. Then $s'_{i,w} = s'_{i,w'}$.*

Proof: Since $\to_i$ is Euclidean and transitive, for any w'', $w \to_i w''$ iff $w' \to_i w''$. By the definition of $R_{A,\psi}(s)$, $s'_{i,w} = s'_{i,w'}$. ■

Definition 21 We say a state formula φ is valid if for all ECGS $\mathcal{G}$, we have $\mathcal{G} \vDash \varphi$, meaning w^0, fus $\vDash \varphi$, where w^0 is the initial state , and fus is the full uniform strategy space.

Example 2 cont'd. We consider whether $\mathcal{G}$ satisfies the following three formulas:

1. $B_1\langle\langle 1,2\rangle\rangle\langle\top\rangle pos_1$, meaning that agent 1 believes that agents 1 and 2 have a strategy to bring the carriage to pos_1 in the next state;
2. $\langle\langle 1,2\rangle\rangle\hat{B}_1\langle\top\rangle pos_1$, meaning that agents 1 and 2 have a strategy to ensure that agent 1 believes that the carriage is possibly in pos_1 in the next state;
3. $((1,2))\langle\top\rangle pos_1$, which is the formula we have considered for Example 1 and does not hold there.

Since both goals $\langle\top\rangle pos_1$ and $\hat{B}_1\langle\top\rangle pos_1$ only concern the next state, at each state, each agent has two strategies: $push$ and $wait$. Figure 3 shows at each state, whether each joint strategy can achieve the goal $\langle\top\rangle pos_1$.

Since $q_0 \to_1 q_2$ and q_2, fus $\vDash \langle\langle 1,2\rangle\rangle\langle\top\rangle pos_1$, we get q_0, fus $\vDash B_1\langle\langle 1,2\rangle\rangle\langle\top\rangle pos_1$.

To check if q_0, fus $\vDash \langle\langle 1,2\rangle\rangle\hat{B}_1\langle\top\rangle pos_1$, we consider if $q_0, (w,w)$, fus $\vDash \hat{B}_1\langle\top\rangle pos_1$, *i.e.*, at q_0, when both agents do w, does 1 believe it possible that $\langle\top\rangle pos_1$? Since agent 1 mistakes q_0 as q_2, and believes it possible that 2 does p, we consider if $q_2, (w,p)$, fus $\vDash \langle\top\rangle pos_1$, which holds. Thus q_0, fus $\vDash \langle\langle 1,2\rangle\rangle\hat{B}_1\langle\top\rangle pos_1$.

Figure 4 shows the procedure of iterative EDS for coalition $\{1,2\}$ and goal $\langle\top\rangle pos_1$. For example, Figure 3 shows at q_2 for agent 1, w dominates p. Thus at q_0 and q_2, agent 1 believes w dominates p, so p is eliminated. Since agent 1 has complete information about q_1, and at q_1, for agent 1, p and w are incomparable, thus no strategy is eliminated. We now move to Round 2. At q_1, 1 compares her two kept strategies p and w. Since agent 1 has complete information about q_1, and at q_1, 2 keeps both p and w, so for agent 1, p and w remains incomparable, thus no strategy is eliminated. It turns out at round 2, no strategy can be eliminated.

Now at q_0, 1 keeps p, and 2 keeps p and w. By the first table of Figure 3, q_0, fus $\nvDash ((1,2))\langle\top\rangle pos_1$. This contrasts to q_0, fs$(q_0) \vDash ((1,2))\langle\top\rangle pos_1$ in JAADL in Example 1. The reason is that at q_0, agent 1's beliefs about domination of strategies do not match the truth, neither do agent 2's beliefs.

4 More Examples

In this section, we illustrate the syntax and semantics of EJAADL with two examples from [11]. Our presentation of the examples differs from theirs in that they give incomplete specifications of agents' beliefs, while we give complete specifications in terms of Kripke models so that we can give the complete processes of iterative elimination of dominated strategies. Moreover, we consider not only joint abilities, but also beliefs about joint abilities.

In both examples, there are two agents 1 and 2, a single atom F. Agent 1 acts first and then agent 2 acts. Each agent can do one of two actions: a and a'. Both actions are public and can always be executed. The atom F is unaffected by any action. The goals concern what agent 1 does first and then what agent 2 does.

Example 3 (Example 2 from [11]) The initial beliefs of the agents are as shown in Figure 5. For example, 1 correctly believes that F is true, 2 does not believe that F is true, neither does 2 believe that F is false, but 2 believes that 1 believes that F is true or 1 believes that F is false. The goal is: if F is true, 1 does a and 2 doesn't do a, otherwise 1 does not do a and 2 does a (see Figure 6). This goal can be represented with the following EJAADL formula

$$\psi_1 : F \wedge [\neg a_1; \top]\bot \wedge [\top; a_2]\bot \vee \neg F \wedge [a_1; \top]\bot \wedge [\top; \neg a_2]\bot.$$

Intuitively, the two agents can achieve the goal. Since 2 believes that 1 believes that F is true or 1 believes that F is false, 2 will do the opposite of the action of 1, which serves as a signal to 2. Since 1 believes that F is true, 1 will do a. Thus they achieve the goal.

We now give formal analysis of this. Given the goal, at each of the initial states, agent 1 has two strategies: a and a'; and then at each of the following states, agent 2 has 4 strategies: (a,a), (a,a'), (a',a), and (a',a'), where (a_1,a_2) means after seeing a do a_1, and after seeing a' do a_2.

Figure 7 shows at each initial state, whether each joint strategy can achieve the goal. Figure 8 shows the procedure of iterative EDS. At round 1, for example, at v_0, u_0, and x_0, for agent 2, the accessible states are u_0 and x_0, (a,a') dominates all other strategies, which are eliminated. Moving to Round 2, at w_0, 2 keeps (a',a) and (a',a');

	1\2	(a,a)	(a,a')	(a',a)	(a',a')
u_0, v_0, w_0	a	×	×	✓	✓
(F)	a'	×	×	×	×
x_0	a	×	×	×	×
$(\neg F)$	a'	✓	×	✓	×

Figure 7. Payoff matrices for Example 3

Round 1	
$w_0, v_0, 1:$	a
$u_0, 1:$	a
$x_0, 1:$	a'
$w_0, 2:$	$(a',a), (a',a')$
$v_0, u_0, x_0, 2:$	(a',a)

Figure 8. Iterative EDS for Example 3

since 2 has complete information at w_0 and 1 keeps a, (a',a) and (a',a') are equivalent, and are both kept. Thus there is no change at Round 2. By Figure 7, joint ability holds at all states. Thus

$$v_0, \text{fus} \models ((1,2))\psi_1 \wedge B_1((1,2))\psi_1 \wedge B_2((1,2))\psi_1.$$

Example 4 (Example 3 from [11]) The initial beliefs of the agents are the same as in Example 3. The goal is different now: if F is true, 1 does anything and 2 doesn't do a, otherwise 1 does anything and 2 does a (see Figure 6). This goal can be represented with the following EJAADL formula

$$\psi_2 : F \wedge [\top; a_2]\bot \vee \neg F \wedge [\top; \neg a_2]\bot.$$

In a sense, the goal ψ_2 is easier to achieve than the goal ψ_1 in Example 3, since it does not require any specific action from 1. Yet, in this case, there is no joint ability to achieve the goal. Intuitively, 1 can not help 2 figure out what to do.

Figure 9 shows at each initial state, whether each joint strategy can achieve the goal. Figure 10 shows the procedure of iterative EDS. The only change from Round 1 to 2 is that at u_0, 1 eliminates a. This is because: the only accessible state from u_0 is itself; at u_0, 2 only keeps (a,a') and (a',a'); now a' dominates a. The only change from Round 2 to 3 is that at v_0, u_0, and x_0, 2 eliminates (a',a'). This is because: the accessible states from any of these states are u_0 and x_0 and now at u_0, 1 only keeps a'; now (a,a') dominates (a',a'). Going to Round 4, at w_0 and v_0, 1 keeps a and a'; the only accessible world is w_0, at w_0, 2 only keeps (a',a'), so a and a' are equivalent, and are both kept. Similarly, at x_0, 1 keeps both a and a'. Thus there is no change at Round 4. By Figure 9, joint ability holds at all states except v_0. Thus

$$v_0, \text{fus} \models \neg((1,2))\psi_2 \wedge B_1((1,2))\psi_2 \wedge B_2((1,2))\psi_2.$$

5 Properties of EJAADL

In this section, we analyze basic properties of EJAADL, and adapt the sufficient/necessary conditions for joint abilities from JAADL.

The first property shows that if agent i has a strategy to ensure she believes ψ, then any coalition including herself has joint ability to achieve ψ.

	1\2	(a,a)	(a,a')	(a',a)	(a',a')
u_0, v_0, w_0	a	×	×	✓	✓
(F)	a'	×	✓	×	✓
x_0	a	✓	✓	×	×
$(\neg F)$	a'	✓	✓	×	×

Figure 9. Payoff matrices for Example 4

Round 1	
$w_0, v_0, 1:$	a, a'
$u_0, 1:$	a, a'
$x_0, 1:$	a, a'
$w_0, 2:$	(a',a')
$v_0, u_0, x_0, 2:$	(a,a'), (a',a')

Round 2	
$w_0, v_0, 1:$	a, a'
$u_0, 1:$	a'
$x_0, 1:$	a, a'
$w_0, 2:$	(a',a')
$v_0, u_0, x_0, 2:$	(a,a'), (a',a')

Round 3	
$w_0, v_0, 1:$	a, a'
$u_0, 1:$	a'
$x_0, 1:$	a, a'
$w_0, 2:$	(a',a')
$v_0, u_0, x_0, 2:$	(a,a')

Figure 10. Iterative EDS for Example 4

Proposition 2 *$\langle\langle i\rangle\rangle B_i\psi \to ((A))\psi$, $i \in A$, is valid.*

Proof: Let $\mathcal{G}$ be an ECGS, w its initial state. Let $w, \text{fus} \models \langle\langle i\rangle\rangle B_i\psi$. Then there is a strategy $\sigma_i \in \text{fus}_{i,w}$ s.t. for all $\sigma_{-i} \in \text{fus}_{-i,w}$, $w, (\sigma_i, \sigma_{-i}), \text{fus} \models B_i\psi$. By the semantics of $B_i\psi$, for all w' s.t. $w \to_i w'$, for all $\sigma'_{-i} \in \text{fus}_{-i,w'}$, $w', (\sigma_i, \sigma'_{-i}), \text{fus} \models \psi$. So σ_i is a winning strategy at w'. So σ_i is kept at w. Now suppose σ'_i is also kept at w. Then we have for any w' s.t. $w \to_i w'$, σ'_i is a winning strategy at w'. So σ'_i is also a winning strategy at w. So $w, \text{fus} \models ((A))\psi$. ■

It is easy to prove the second property, which states that in order to have joint abilities, any agent in the coalition cannot believe a punishment strategy to be a dominating strategy.

Proposition 3 *$((A))\psi \to (A)_\psi \bigwedge_{i\in A} \neg\langle\langle i\rangle\rangle\neg\psi$ is valid.*

Now we show that two sufficient conditions for joint abilities in JAADL can be modified to hold in EJAADL. Essentially, we add the condition that a strategy is a winning/optimal strategy iff it is believed to be a winning/optimal strategy. The idea of proof is similar to that of Proposition 2.

Theorem 1 *If at stage k, agent i has a winning strategy and a strategy is a winning strategy iff i believes it to be a winning strategy, then there is joint ability at stage $k+1$.*

Proof: Suppose i has a stage k winning strategy σ_i. Then i believes it to be a winning strategy, so σ_i remains at stage $k+1$. For every strategy σ'_i which remains at stage $k+1$, i must believe it to be a winning strategy at stage k, otherwise, i believes σ_i dominates σ'_i, hence it is already eliminated. So σ'_i is a stage k winning strategy. Thus there is joint ability at stage $k+1$. ■

Theorem 2 *If $\langle\langle A\rangle\rangle\psi$ holds, and at stage k, each agent in A has an optimal strategy and a strategy is optimal iff it is believed to be optimal, then there is joint ability at stage $k+1$.*

Proof: Suppose $\langle\langle A\rangle\rangle\psi$ holds, and at stage k, each agent in A has an optimal strategy. Then for each joint strategy σ_{all} where for each $a \in A$, σ_a is an optimal strategy, σ_{all} achieves ψ. Each non-optimal strategy is believed to be non-optimal and will be deleted at the next stage. Hence, there is joint ability at stage $k+1$. ■

Finally, the necessary condition for joint abilities in JAADL can be modified to hold in EJAADL.

Theorem 3 *Suppose at some stage, no agent in A has a winning strategy, and any two strategies are belief-wise either equivalent or incomparable, then there is no joint ability.*

Proof: When any two strategies are belief-wise either equivalent or incomparable, no elimination is possible. Since no agent in A has a winning strategy, there is no joint ability. ■

6 Model Checking Memoryless EJAADL

In this section, we first adapt the algorithm to model-check memoryless JAADL to EJAADL, and show that model checking memoryless EJAADL is in EXPTIME. Then we give a PSPACE algorithm for model-checking the memoryless version of the fragment of EJAADL without the iterated elimination operator.

6.1 Memoryless EJAADL

We first review the main idea of model checking memoryless JAADL from [17]. Recall JAADL path formulas are LDL formulas over JAADL state formulas. The algorithm essentially follows the semantic definition, and is a labeling algorithm which returns the set of all states satisfying a given state formula. When checking a path formula ψ w.r.t. a state w and a joint strategy σ_{all}, we first label each maximal state sub-formula φ of ψ, introduce a fresh atom p_φ to represent φ, and so get a pure LDL formula ψ_{ldl}. The infinite path $out(w, \sigma_{all})$ can be viewed as a finite Kripke model with initial state w, denoted as $Kripke(w, \sigma_{all})$. To verify whether $Kripke(w, \sigma_{all}) \vDash_{ldl} \psi_{ldl}$, we call the LDL model-checking algorithm [10], which is polynomial time in the model size and exponential time in the formula size.

The main idea of adapting the model-checking algorithm from JAADL to EJAADL is as follows. In EJAADL, there is an additional path formula constructor $B_i\psi$. To handle this constructor, we also give a labeling algorithm for path formulas: given a path formula ψ, a state w and a joint strategy σ_{all}, the algorithm returns the set of states satisfying ψ under σ_{all}. We label $B_i\psi$ formulas separately; when labeling a non-$B_i\psi$ path formula, we also introduce a fresh atom for each maximal $B_i\psi$ sub-formula.

We now formally state the model checking problem for memoryless EJAADL: Given an ECGS $\mathcal{G}$, and an EJAADL formula φ, decide if $w^0, \mathit{fmus} \vDash \varphi$, where w^0 is the initial state of $\mathcal{G}$, and *fmus* denotes the full memoryless uniform strategy space of $\mathcal{G}$. We give the whole algorithm where we use the $image$ function: $img(w, \to_i) = \{w' \mid w \to_i w'\}$. $Sub(\varphi)$ denotes the subformulas of φ. To compute $StrS^\infty(A, \psi, w, s)$, we repeat $s \leftarrow StrS(A, \psi, w, s)$ until there is no change to s.

Theorem 4 *Model-checking memoryless EJAADL can be done in time exponential in the model size and formula size.*

Proof: Let n be the model size and l the formula size. Then the number of different joint memoryless uniform strategies is $O(2^n)$. The computationally demanding parts of the algorithm are the evaluation of the $\langle\langle A\rangle\rangle\psi$ and $B_i\psi$ operators, the calculation of the reduction and iterative reduction of a strategy space, and calling LDL model-checking. Each of these parts takes time exponential in either the model size or the formula size. Thus, by induction on the size of the formula, the whole algorithm takes time $O(2^{nl})$. ■

Algorithm 1: Labeling State Formulas

function $Label(\mathcal{G}, s, \varphi)$
foreach φ' *in* $Sub(\varphi)$ **do**
  case $\varphi' = p$ **do** $[\varphi']_s \leftarrow \{w \in W \mid p \in L(w)\}$;
  case $\varphi' = \neg\varphi$ **do** $[\varphi']_s \leftarrow W - [\varphi]_s$;
  case $\varphi' = \varphi_1 \wedge \varphi_2$ **do** $[\varphi']_s \leftarrow [\varphi_1]_s \cap [\varphi_2]_s$;
  case $\varphi' = \langle\langle A\rangle\rangle\psi$ **do**
    $[\varphi']_s \leftarrow \{w \mid \exists\sigma_A \in s_{A,w} \forall\sigma_{-A} \in s_{-A,w}.$
      $w \in PathL(\mathcal{G}, (\sigma_A, \sigma_{-A}), s, \psi)\}$;
  case $\varphi' = (A)_\psi\varphi_1$ **do** $[\varphi']_s \leftarrow [\varphi_1]_{StrS(A,\psi,s)}$;
  case $\varphi' = (A)^\infty_\psi\varphi_1$ **do** $[\varphi']_s \leftarrow [\varphi_1]_{StrS^\infty(A,\psi,s)}$;
  case $\varphi' = B_i\varphi_1$ **do** $[\varphi']_s \leftarrow \{w \mid img(w, \to_i) \subseteq [\varphi_1]_s\}$;
return $[\varphi]_s$

Algorithm 2: Labeling Path Formulas

function $PathL(\mathcal{G}, \sigma_{all}, s, \psi)$
if $\psi = B_i\psi'$ **then**
  return $\{w \mid \forall w' \in img(w, \to_i) \forall\sigma'_{-i} \in s_{-i,w'}.$
    $w' \in PathL(\mathcal{G}, (\sigma_i, \sigma'_{-i}), s, \psi')\}$
Max(ψ) ← the set of maximal state or $B_i\psi'$ subformulas in ψ;
foreach $\phi \in Max(\psi)$ **do**
  define a fresh atom p_ϕ;
  let $p_\phi \in L(w)$ iff $w \in Label(\mathcal{G}, s, \phi)$ or
   $w \in PathL(\mathcal{G}, \sigma_{all}, s, \phi)$;
replace each occurrence in ψ of $\phi \in$ Max(ψ) by p_ϕ to get a pure LDL formula ψ_{ldl};
return $\{w \in W \mid Kripke(w, \sigma_{all}) \vDash_{ldl} \psi_{ldl}\}$

6.2 Memoryless Infinity-free EJAADL

We are interested in exploring fragments of EJAADL with lower computational complexity. Note that when joint abilities hold, we often have low-order joint abilities, *i.e.*, after a few round of elimination of dominated strategies, any joint strategy of the coalition achieves the goal. This motivates us to consider the infinity-free fragment of EJAADL, *i.e.*, the fragment without the iterated elimination operator $(A)^\infty_\psi$. We denote this fragment by EJAADL$^-$.

In the following, we give a PSPACE algorithm for model-checking memoryless EJAADL$^-$. The main idea of the algorithm is as follows: Let $R(\mathit{fmus}, \oplus_1 \cdots \oplus_k)$ denote the strategy space obtained from

Algorithm 3: Calculating Reduced Strategy Space

function $StrS(A, \psi, s)$
foreach $w \in W$ **do**
  foreach $i \in A$ **do**
    foreach $\sigma_i \in s_{i,w}$ *and* $w' \in img(w, \to_i)$ **do**
      compute $M_{\psi,w',s}(\sigma_i)$, *i.e.*,
       $\{\sigma_{-i} \in s_{-i,w'} \mid w' \in PathL(\mathcal{G}, (\sigma_i, \sigma_{-i}), s, \psi)\}$
    foreach $\sigma_i, \sigma'_i \in s_{i,w}$ **do**
      if *for all* $w' \in img(w, \to_i)$,
       $M_{\psi,w',s}(\sigma_i) \supseteq M_{\psi,w',s}(\sigma'_i)$ *and there is*
       $w' \in img(w, \to_i)$ *s.t.*
       $M_{\psi,w',s}(\sigma_i) \supset M_{\psi,w',s}(\sigma'_i)$ **then**
        $s_{i,w} \leftarrow s_{i,w} - \{\sigma'_i\}$;
return s

$fmus$ by applying a sequence $\oplus_1 \cdots \oplus_k$ of elimination operators of the form $(A)_\psi$. Instead of using exponential space to explicitly store $R(fmus, \oplus_1 \cdots \oplus_k)$, we just store $\oplus_1 \cdots \oplus_k$. When we need a strategy from $R(fmus, \oplus_1 \cdots \oplus_k)$, we enumerate strategies σ_i from $fmus$ in lexicographical order, and check if it is in $R(fmus, \oplus_1 \cdots \oplus_k)$, which is done recursively by checking if it is in $R(fmus, \oplus_1 \cdots \oplus_{k-1})$ and retained after applying $\oplus_k$. Let σ_A be a group uniform strategy of A. We use $NextU(\sigma_A)$ to represent the next group uniform strategy of A in lexicographical order.

There are three functions. Function $MLabel(\mathcal{G}, \oplus_1 \cdots \oplus_k, \varphi)$ returns the set of states satisfying φ w.r.t. the strategy space $R(fmus, \oplus_1 \cdots \oplus_k)$. Given an ECGS $\mathcal{G}$, and an EJAADL$^-$ formula φ, to decide if $w^0, fmus \models \varphi$, we check if $w^0 \in MLabel(\mathcal{G}, nil, \varphi)$. Function $MPathL(\mathcal{G}, \sigma_{all}, \oplus_1 \cdots \oplus_k, \psi)$ returns the set of states where the unique path determined by σ_{all} satisfies ψ w.r.t. $R(fmus, \oplus_1 \cdots \oplus_k)$. Function $Keep(\mathcal{G}, w, \sigma_i, \oplus_1 \cdots \oplus_k)$ determines if $\sigma_i \in s_{i,w}$ where $s = R(fmus, \oplus_1 \cdots \oplus_k)$. Finally, we use $Keep(\mathcal{G}, w, \sigma_A, \oplus_1 \cdots \oplus_k)$ for $\bigwedge_{i \in A} Keep(\mathcal{G}, w, \sigma_i, \oplus_1 \cdots \oplus_k)$.

Algorithm 4: Labeling EJAADL$^-$ State Formulas

function $MLabel(\mathcal{G}, \oplus_1 \cdots \oplus_k, \varphi)$,
  where φ is in the scope of $\oplus_1 \cdots \oplus_k$
foreach φ' *in* $Sub(\varphi)$ **do**
  case $\varphi' = p$ **do** $[\varphi'](\oplus_1 \cdots \oplus_k) \leftarrow \{w \in W \mid p \in L(w)\}$;
  case $\varphi' = \neg\varphi$ **do** $[\varphi'](\oplus_1 \cdots \oplus_k) \leftarrow W - [\varphi](\oplus_1 \cdots \oplus_k)$;
  case $\varphi' = \varphi_1 \wedge \varphi_2$ **do**
    $[\varphi'](\oplus_1 \cdots \oplus_k) \leftarrow [\varphi_1](\oplus_1 \cdots \oplus_k) \cap [\varphi_2](\oplus_1 \cdots \oplus_k)$;
  case $\varphi' = \langle\langle A \rangle\rangle \psi$ **do**
    $[\varphi'](\oplus_1 \cdots \oplus_k) \leftarrow \emptyset$;
    foreach $w \in W$ **do**
      if *there is* σ_A *enumerated by* $NextU$ *s.t.* $Keep(\mathcal{G}, w, \sigma_A, \oplus_1 \cdots \oplus_k) = true$ *and for all* σ_{-A} *enumerated by* $NextU$ *s.t.* $Keep(\mathcal{G}, w, \sigma_{-A}, \oplus_1 \cdots \oplus_k) = true$, *we have* $w \in MPathL(\mathcal{G}, (\sigma_A, \sigma_{-A}), \oplus_1 \cdots \oplus_k, \psi)$ **then**
        $[\varphi'](\oplus_1 \cdots \oplus_k) \leftarrow [\varphi'](\oplus_1 \cdots \oplus_k) \cup \{w\}$;
  case $\varphi' = (A)_\psi \varphi_1$ **do**
    $[\varphi'](\oplus_1 \cdots \oplus_k) \leftarrow [\varphi_1](\oplus_1 \cdots \oplus_k\ (A)_\psi)$;
  case $\varphi' = B_i \varphi_1$ **do**
    $[\varphi'](\oplus_1 \cdots \oplus_k) \leftarrow$
      $\{w \mid img(w, \rightarrow_i) \subseteq [\varphi_1](\oplus_1 \cdots \oplus_k)\}$;
return $[\varphi](\oplus_1 \cdots \oplus_k)$

Theorem 5 *Model-checking memoryless infinity-free EJAADL can be done in polynomial space in both the model size and formula size.*

Proof: A single $MLabel(\mathcal{G}, \oplus_1 \cdots \oplus_k, \varphi)$ function call (excluding recursions) requires polynomial space w.r.t. the model size and formula size to maintain its parameters, a group strategy, and state space. Similarly, single function calls about $MPathL(\mathcal{G}, \sigma_{all}, \oplus_1 \cdots \oplus_k, \psi)$ and $Keep(\mathcal{G}, w, \sigma_i, \oplus_1 \cdots \oplus_k)$ require polynomial space too. Here note that the LDL model-checking algorithm takes polynomial space in the model size and formula size. In each recursive call about any one of the three functions, either formulas are decomposed, or operator sequences are reduced, the recursion depth cannot be more than polynomial size about the size of model and formula. So the whole procedure $MLabel(\mathcal{G}, nil, \varphi)$ requires only polynomial space. ■

Algorithm 5: Labeling EJAADL$^-$ Path Formulas

function $MPathL(\mathcal{G}, \sigma_{all}, \oplus_1 \cdots \oplus_k, \psi)$
  where ψ is in the scope of $\oplus_1 \cdots \oplus_k$
if $\psi = B_i \psi'$ **then**
  $S \leftarrow \emptyset$;
  foreach $w \in W$ **do**
    if *for* $w' \in img(w, \rightarrow_i)$, *for all* σ'_{-i} *enumerated by* $NextU$ *s.t.* $Keep(\mathcal{G}, w', \sigma'_{-i}, \oplus_1 \cdots \oplus_k) = true$, *we have* $w' \in MPathL(\mathcal{G}, (\sigma_i, \sigma'_{-i}), \oplus_1 \cdots \oplus_k, \psi')$ **then**
      $S \leftarrow S \cup \{w\}$;
  return S;
Max$(\psi) \leftarrow$ the set of maximal state or $B_i \psi'$ subformulas in ψ;
foreach $\phi \in Max(\psi)$ **do**
  define a fresh atom p_ϕ;
  let $p_\phi \in L(w)$ iff $w \in MLabel(\mathcal{G}, \oplus_1 \cdots \oplus_k, \phi)$ or $w \in MPathL(\mathcal{G}, \sigma_{all}, \oplus_1 \cdots \oplus_k, \phi)$;
replace each occurrence in ψ of $\phi \in$ Max(ψ) by p_ϕ to get a pure LDL formula ψ_{ldl};
return $\{w \in W \mid Kripke(w, \sigma_{all}) \models_{ldl} \psi_{ldl}\}$

Algorithm 6: Checking Strategy Retaining

function $Keep(\mathcal{G}, w, \sigma_i, \oplus_1 \cdots \oplus_k)$
if $k = 0$ **then return** *true*;
let $\oplus_k = (A)_\psi$;
if $i \notin A$ **then**
  return $Keep(\mathcal{G}, w, \sigma_i, \oplus_1 \cdots \oplus_{k-1})$;
if $Keep(\mathcal{G}, w, \sigma_i, \oplus_1 \cdots \oplus_{k-1}) = false$ **then**
  return *false*;
if *there is* σ'_i *enumerated by* $NextU$ *s.t.* $Keep(\mathcal{G}, w, \sigma'_i, \oplus_1 \cdots \oplus_{k-1}) = true$ **then**
  if *for all* $w' \in img(w, \rightarrow_i)$ *we have* $w' \in MPathL(\mathcal{G}, (\sigma'_i, \sigma_{-i}), \oplus_1 \cdots \oplus_{k-1}, \psi)$ *when* $w' \in MPathL(\mathcal{G}, (\sigma_i, \sigma_{-i}), \oplus_1 \cdots \oplus_{k-1}, \psi)$,
  and, there exists $w' \in img(w, \rightarrow_i)$ *s.t.* $w' \in MPathL(\mathcal{G}, (\sigma'_i, \sigma_{-i}), \oplus_1 \cdots \oplus_{k-1}, \psi)$ *but* $w' \notin MPathL(\mathcal{G}, (\sigma_i, \sigma_{-i}), \oplus_1 \cdots \oplus_{k-1}, \psi)$ **then**
    return *false*;
return *true*;

7 Conclusions

In this paper, by extending JAADL, we propose a modal logic EJAADL for joint abilities for imperfect information games where agents may have false beliefs about the world. The main idea is that elimination of dominated strategies is now based on beliefs about the world, rather than facts about the world. In comparison to Ghaderi *et al.*'s work, which uses the expressive first-order language of situation calculus to formalize joint abilities, our contributions are as follows: we give a clean semantics for EJAADL, illustrate it with several examples by giving the complete processes of iterative elimination of dominated strategies, and finally give an EXPTIME algorithm for model-checking memoryless EJADDL and a PSPACE algorithm for memoryless infinity-free EJAADL. In the future, we are interested in extending EJAADL by specifying for each agent the set of agents whose strategy she is informed of, following [3].

Acknowledgements

We thank Liping Xiong for his great help with the PSPACE model-checking algorithm in the paper. This work received support from the Natural Science Foundation of China under Grant No. 62076261.

References

[1] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman, 'Alternating-time temporal logic', *J. ACM*, **49**(5), 672–713, (2002).

[2] Francesco Belardinelli, 'A logic of knowledge and strategies with imperfect information', *LAMAS*, **15**, 1–15, (2015).

[3] Francesco Belardinelli, Sophia Knight, Alessio Lomuscio, Bastien Maubert, Aniello Murano, and Sasha Rubin, 'Reasoning About Agents That May Know Other Agents' Strategies', in *IJCAI*, pp. 1787–1793, (2021).

[4] Francesco Belardinelli, Alessio Lomuscio, Aniello Murano, and Sasha Rubin, 'Verification of Broadcasting Multi-Agent Systems against an Epistemic Strategy Logic', in *IJCAI*, pp. 91–97, (2017).

[5] Francesco Belardinelli, Alessio Lomuscio, Aniello Murano, and Sasha Rubin, 'Verification of Multi-agent Systems with Imperfect Information and Public Actions', in *AAMAS*, pp. 1268–1276, (2017).

[6] Raphaël Berthon, Bastien Maubert, Aniello Murano, Sasha Rubin, and Moshe Y. Vardi, 'Strategy logic with imperfect information', in *LICS*, pp. 1–12, (2017).

[7] N. Bulling, J. Dix, and W. Jamroga, 'Model Checking Logics of Strategic Ability: Complexity*', in *Specification and Verification of Multi-agent Systems*, 125–159, Springer, Boston, MA, (2010).

[8] Petr Cermák, Alessio Lomuscio, Fabio Mogavero, and Aniello Murano, 'MCMAS-SLK: A Model Checker for the Verification of Strategy Logic Specifications', *CoRR*, **abs/1402.2948**, (2014).

[9] Catalin Dima, Constantin Enea, and Dimitar P. Guelev, 'Model-Checking an Alternating-time Temporal Logic with Knowledge, Imperfect Information, Perfect Recall and Communicating Coalitions', volume 25 of *EPTCS*, pp. 103–117, (2010).

[10] Peter Faymonville and Martin Zimmermann, 'Parametric Linear Dynamic Logic', *Information and Computation*, **253**, 237–256, (2017).

[11] Hojjat Ghaderi, Hector J. Levesque, and Yves Lespérance, 'A Logical Theory of Coordination and Joint Ability', in *AAAI*, pp. 421–426, (2007).

[12] Peter Hawke, 'The Logic of Joint Ability in two-Player Tacit Games', *Rev. Symb. Logic*, **10**(3), 481–508, (2017).

[13] Wiebe van der Hoek and Michael J. Wooldridge, 'Cooperation, Knowledge, and Time: Alternating-time Temporal Epistemic Logic and its Applications', *Studia Logica*, **75**(1), 125–157, (2003).

[14] Wojciech Jamroga and Wiebe van der Hoek, 'Agents That Know How to Play', *Fundam. Inf.*, **63**(2–3), 185–219, (2004).

[15] Wojciech Jamroga and Thomas Ågotnes, 'Constructive knowledge: what agents can achieve under imperfect information', *J. Appl. Non Class. Logics*, **17**(4), 423–475, (2007).

[16] Guifei Jiang, Dongmo Zhang, Laurent Perrussel, and Heng Zhang, 'Epistemic GDL: A logic for representing and reasoning about imperfect information games', *Artif. Intell.*, **294**, 103453, (2021).

[17] Zhaoshuai Liu, Liping Xiong, Yongmei Liu, Yves Lespérance, Ronghai Xu, and Hongyi Shi, 'A Modal Logic for Joint Abilities under Strategy Commitments', in *IJCAI*, pp. 1805–1812, (2020).

[18] Bastien Maubert and Aniello Murano, 'Reasoning about Knowledge and Strategies under Hierarchical Information', in *KR*, pp. 530–540, (2018).

[19] Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe Y. Vardi, 'Reasoning About Strategies: On the Model-Checking Problem', *ACM Trans. Comput. Log.*, **15**(4), 34:1–34:47, (2014).

[20] Martin J. Osborne and Ariel Rubinstein, *A Course in Game Theory*, The MIT Press, 1999.

[21] Marc Pauly, 'A Modal Logic for Coalitional Power in Games', *J. Log. Comput.*, **12**(1), 149–166, (2002).

[22] Raymond Reiter, *Knowledge in action: logical foundations for specifying and implementing dynamical systems*, MIT press, 2001.

[23] Moshe Y Vardi, 'The rise and fall of linear temporal logic', *Proceedings of the 2nd International Syposium on Games, Automata, Logics, and Formal Verification (11)*, (2011).